ℓ_P MINIMIZATION FOR SPARSE VECTOR RECONSTRUCTION

N. Mourad and James P. Reilly

Department of Electrical & Computer Eng.
McMaster University, 1280 Main St. W., Hamilton, Ontario, Canada L8S 4K1
email: mouradna@mcmaster.ca, reillyj@mcmaster.ca

ABSTRACT

In this paper we present a new technique for minimizing a class of nonconvex functions for solving the problem of under–determined systems of linear equations. The proposed technique is based on locally replacing the nonconvex objective function by a convex objective function. The main property of the utilized convex function is that it is minimized at a point that reduces the original concave function. The resulting algorithm is iterative and outperforms some previous algorithms that have been applied to the same problem .

Indexing Terms: *compressed sensing, sparse component analysis, optimization.* .

I. INTRODUCTION

The problem of finding a unique solution of linear under–determined systems of equations $\boldsymbol{As} = \boldsymbol{x}$, where $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and $m < n$, is challenging indeed. Since the number of equations is less than the number of unknowns, it is known that such systems have infinitely many solutions and thus, it is to identify which of these candidate solutions is the desired one, it is necessary to impose constraints on the candidate solution.

A powerful constraint that can be utilized is the "*sparsity*" of the solution vector. Sparsity means that the solution vector has few nonzero elements. This constraint is realistic in many situations, e.g., signal representation using a small number of atoms from an overcomplete dictionary [1], extraction of the event related potential (ERP) signal from a background electroencephalographic (EEG) noise [2], and direction of arrival (DOA) estimation[3].

There are many algorithms developed for solving the above mentioned problem. Most of these algorithms are based on solving the following general optimization problem

$$\hat{\boldsymbol{s}} = \arg\min_{\boldsymbol{s}} g(\boldsymbol{s}) \quad \text{subject to} \quad \boldsymbol{x} = \boldsymbol{As} \tag{1}$$

where $g(\cdot)$ is an objective function that measures the *diversity* (antisparsity) of the solution vector.

Maximizing the sparsity of the solution vector can be achieved by selecting $g(\boldsymbol{s}) = ||\boldsymbol{s}||_0$, where the function $||\boldsymbol{s}||_0$ simply counts the number of nonzeros in $\boldsymbol{s}$. Unfortunately, until now, an effective algorithm for solving this optimization problem did not exist [4]. On the other hand, it is known that selecting $g(\boldsymbol{s}) = ||\boldsymbol{s}||_2$ produces a nonsparse solution [5].

Over the last two decades, $g(\boldsymbol{s}) = ||\boldsymbol{s}||_1$ was extensively utilized for measuring the diversity of the solution vector. The resulting optimization problem is convex and can be solved efficiently [6]. More interestingly, under certain conditions on the dictionary matrix $\boldsymbol{A}$, the solution obtained by minimizing $||\boldsymbol{s}||_1$ is exactly the same as that obtained by minimizing $||\boldsymbol{s}||_0$ [7]. Unfortunately, these conditions are somewhat restrictive.

In [8] it was shown that, minimizing $||\boldsymbol{s}||_p^p$ for $0 < p < 1$, can perform much better than minimizing $||\boldsymbol{s}||_1$ in the sense that a minimum number of observations are needed for exact reconstruction of the sparse solution vector.

In this paper, we propose a new algorithm for solving the problem of under–determined systems of equations when the solution vector is known to be sparse. This algorithm is based on iteratively minimizing a concave objective function that measures the *diversity* (antisparsity) of the solution vector. Since minimizing a convex function is more efficient than minimizing a concave function, the proposed algorithm is based on locally replacing the concave objective function by a convex function. The key point is to iteratively select the convex function properly such that it will be minimized at a point that reduces the original concave function. The algorithm will thus converge to at least a local minimum.

Since the proposed algorithm is based on **M**inimizing a **C**oncave function via a **C**onvex function **A**pproximation, we will refer to the derived algorithm as MCCA.

The paper is organized as follows. In Section II the proposed local convex function is presented. The MCCA algorithm is derived in Section III . Section IV presents some computer simulations for assessing the performance of the proposed algorithms. Finally, conclusions are given in Section V.

II. PROPOSED CONVEX FUNCTION

The class of non–convex (concave) objective functions, $g_c(t)$, considered in this paper have the following property.

Property 1: The function $g_c(|t|)$ is concave and monotonically increasing (decreasing) function for all values of $t \geq 0$ ($t < 0$), respectively.

Examples of such functions which can be used for sparse vector reconstruction are $g_p(t) = |t|^p$ and $g_{log}(t) = log(|t|)$.

The main difficulties associated with minimizing a concave function are: 1) minimizing a concave function is not as

This work was supported in part by the Egyptian Ministry of High Education and the Natural Sciences and Engineering Research Council of Canada (NSERC).

efficient as minimizing a convex function, and 2) minimizing a non-convex function is associated with multiple local minima in addition to the global minima, while a convex function has only one global minimum.

In this paper, we propose an efficient technique that can be utilized in minimizing non-convex objective functions that obey Property 1. The main idea behind the proposed technique is at each iteration, we locally replace the concave function by a convex function that is minimized at a point where the concave function is reduced in value.

The minimization problem at hand can be stated as follows: starting from an initial point t_0, find t_1 such that $g_c(t_1) \leq g_c(t_0)$. The solution of this problem is provided in the following theorem.

Theorem 1: Consider a monotonically increasing concave function $g_c(t)$ defined for all values of $t \geq 0$, and a starting point t_0. A new point t_1, such that $g_c(t_1) \leq g_c(t_0)$, can be estimated as a minimizer of the following function

$$t_1 = \arg\min_t f(t) = g_c(t_0) + g_c'(t_0)(t-t_0) - 0.5g_c''(t_0)(t-t_0)^2 \quad (2)$$

where $g_c'(t_0)$ and $g_c''(t_0)$ are the gradient and Hessian of $g_c(t)$ at $t = t_0$, respectively.

Proof: Since $g_c(t)$ is concave, $g_c''(t)$ is negative for all values of $t \geq 0$. Consequently, $f''(t)$, the Hessian of $f(t)$, is positive for all values of $t \geq 0$, i.e. $f(t)$ is a convex function. The convexity of $f(t)$ implies that it is always above any tangent, while the concavity of $g_c(t)$ implies that it is always below any tangent. Since $f(t)$ and $g_c(t)$ have a common tangent at $t = t_0$, which is given by the first two terms in (2), $f(t)$ and $g_c(t)$ do not intersect at any other point. Moreover, since $g_c(t)$ is a monotonically increasing function, its tangent at any point, including t_0, always has a positive slope. For the convex function $f(t)$, a tangent with positive slope means that the minimum of this function occurs at a point $t_1 < t_0$. And since $g_c(t)$ is monotonically increasing, this means that $g_c(t_1) < g_c(t_0)$. □

Consider the following corollary.

Corollary 1: A monotonically decreasing concave function $g_c(t)$, defined for all values of $t \leq 0$, can be minimized if it is locally replaced by a convex function $f(t)$, defined in Theorem 1.

Proof: The proof follows readily from the proof of Theorem 1.

For the n dimensional case, $f(\boldsymbol{s})$ can be expressed as

$$\begin{aligned} f(\boldsymbol{s}) &= g(\boldsymbol{s}_0) + (\boldsymbol{s}-\boldsymbol{s}_0)^T \nabla g(\boldsymbol{s}_0) \\ &- 0.5(\boldsymbol{s}-\boldsymbol{s}_0)^T \nabla^2 g(\boldsymbol{s}_0)(\boldsymbol{s}-\boldsymbol{s}_0) \end{aligned} \quad (3)$$

where $\boldsymbol{s} \in \mathbb{R}^n$, $g(\boldsymbol{s}) = \sum_i g_c(s[i])$ for some $g_c(s[i])$ which obeys Property 1, and $\nabla g(\boldsymbol{s}_0)$ and $\nabla^2 g(\boldsymbol{s}_0)$ are the gradient and Hessian of $g(\boldsymbol{s})$ at $\boldsymbol{s} = \boldsymbol{s}_0$, respectively.

Corollary 2: A local minimum of the function $g_p(\boldsymbol{s}) \triangleq \sum_i |s[i]|^p$ under the constraint $\boldsymbol{A}\boldsymbol{s} = \boldsymbol{x}$, where $0 < p < 1$, can be obtained by iteratively minimizing the convex function $f(\boldsymbol{s})$ given by (3), with $g(\boldsymbol{s})$ replaced by $g_p(\boldsymbol{s})$

The proof makes use of the fact that $g_p(t)$ is non-convex and monotonically increasing for $0 < p < 1$. The proof is straightforward but is omitted due to lack of space. □

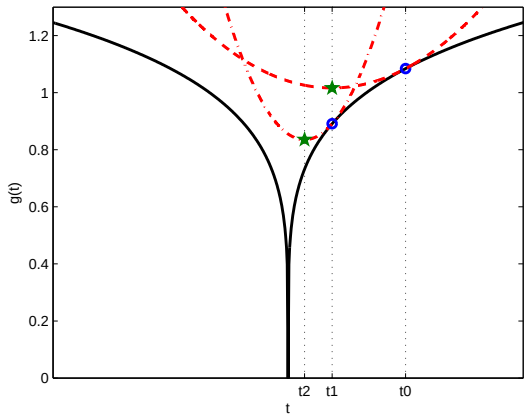

Fig. 1. Demonstration of the proposed iterative technique. The solid curve is $g_c(t)$ whereas the dashed lines represent $f(t)$ at the first two iterations.

The iteration steps are shown in Figure 1. The iterative procedure is repeated until $g_p(t)$ converges to a minima at which $g_p(t_k) = g_p(t_{k-1})$, where k is the iteration index.

Although any other convex function may be utilized instead of the proposed function, the proposed technique is a simple and straightforward way of accomplishing our goals. In the next section, an iterative algorithm that minimizes (1) after replacing $g(\boldsymbol{s})$ by its local convex approximation $f(\boldsymbol{s})$ is derived.

III. THE MCCA ALGORITHM

As described in the previous section, MCCA algorithm is based on replacing $g(\boldsymbol{s})$ in (1) by the convex function $f(\boldsymbol{s})$, defined in (3). Assuming that a starting point $\boldsymbol{s}_0$ is given, then $\boldsymbol{s}_1$, the point at which $g(\boldsymbol{s}_1) \leq g(\boldsymbol{s}_0)$, can be obtained by minimizing the following objective function

$$\begin{aligned} \boldsymbol{s}_1 = \arg\min_{\boldsymbol{s}} \quad & g(\boldsymbol{s}_0) + (\boldsymbol{s}-\boldsymbol{s}_0)^T \nabla g(\boldsymbol{s}_0) \\ & -0.5(\boldsymbol{s}-\boldsymbol{s}_0)^T \nabla^2 g(\boldsymbol{s}_0)(\boldsymbol{s}-\boldsymbol{s}_0) \\ \text{subject to} \quad & \boldsymbol{x} = \boldsymbol{A}\boldsymbol{s}. \end{aligned} \quad (4)$$

This optimization problem can be solved by following the standard method of Lagrangian multipliers (see, e.g. [6]). We define the Lagrangian $L(\boldsymbol{s}, \boldsymbol{\lambda})$ as

$$\begin{aligned} L(\boldsymbol{s}, \boldsymbol{\lambda}) &= g(\boldsymbol{s}_0) + (\boldsymbol{s}-\boldsymbol{s}_0)^T \nabla g(\boldsymbol{s}_0) \\ &- 0.5(\boldsymbol{s}-\boldsymbol{s}_0)^T \nabla^2 g(\boldsymbol{s}_0)(\boldsymbol{s}-\boldsymbol{s}_0) + \boldsymbol{\lambda}^T(\boldsymbol{A}\boldsymbol{s}-\boldsymbol{x}) \end{aligned} \quad (5)$$

where $\boldsymbol{\lambda}$ is the $(m \times 1)$ vector of Lagrange multipliers. A necessary condition for the Lagrangian $L(\boldsymbol{s}, \boldsymbol{\lambda})$ to be minimized at $\boldsymbol{s}_1$ is that $(\boldsymbol{s}_1, \boldsymbol{\lambda}_*)$ be stationary points of the Lagrangian function, i.e.

$$\begin{aligned} \nabla_{\boldsymbol{s}} L(\boldsymbol{s}_1, \boldsymbol{\lambda}_*) &= \nabla g(\boldsymbol{s}_0) - \nabla^2 g(\boldsymbol{s}_0)(\boldsymbol{s}_1 - \boldsymbol{s}_0) \\ &\quad + \boldsymbol{A}^T \boldsymbol{\lambda}_* = 0, \end{aligned} \quad (6)$$

$$\nabla_{\boldsymbol{\lambda}} L(\boldsymbol{s}_1, \boldsymbol{\lambda}_*) = \boldsymbol{A}\boldsymbol{s}_1 - \boldsymbol{x} = 0. \quad (7)$$

For convenience we use $\boldsymbol{d}$ and $\boldsymbol{H}$ to refer to the gradient $\nabla g(\boldsymbol{s}_0)$ and the inverse of the Hessian $(\nabla^2 g(\boldsymbol{s}_0))^{-1}$, respectively. In doing so, we assume that the Hessian matrix

is invertible. As will be seen later, this assumption is always true for the cases considered in this paper. From (6) we have

$$\boldsymbol{s}_1 = \boldsymbol{s}_0 + \boldsymbol{H}(\boldsymbol{d} + \boldsymbol{A}^T\boldsymbol{\lambda}_*). \quad (8)$$

Substituting this equation into (7), and solving for $\boldsymbol{\lambda}_*$ we get

$$\begin{aligned} \boldsymbol{\lambda}_* &= (\boldsymbol{AHA}^T)^{-1}(\boldsymbol{x} - \boldsymbol{As}_0 - \boldsymbol{AHd}) \\ &= -(\boldsymbol{AHA}^T)^{-1}\boldsymbol{AHd} \end{aligned} \quad (9)$$

where the second equality holds because $\boldsymbol{s}_0$ is feasible, i.e. $\boldsymbol{As}_0 = \boldsymbol{x}$. Substituting (9) back into (8), we get the following general expression for $\boldsymbol{s}_1$:

$$\boldsymbol{s}_1 = \boldsymbol{s}_0 + \boldsymbol{Hd} - \boldsymbol{HA}^T(\boldsymbol{AHA}^T)^{-1}\boldsymbol{AHd}.$$

More generally, the estimated value of the solution at the kth iteration can be expressed as

$$\boldsymbol{s}_k = \boldsymbol{s}_{k-1} + \boldsymbol{H}_k\boldsymbol{d}_k - \boldsymbol{H}_k\boldsymbol{A}^T(\boldsymbol{AH}_k\boldsymbol{A}^T)^{-1}\boldsymbol{AH}_k\boldsymbol{d}_k \quad (10)$$

where both $\boldsymbol{H}_k$ and $\boldsymbol{d}_k$ are calculated at $\boldsymbol{s}_{k-1}$.

III-A. $g_p(\boldsymbol{s})$ minimization

When $g_p(\boldsymbol{s}) = \sum_i |s[i]|^p$, where $0 < p < 1$, is utilized as an objective function in (1), the ith element of the gradient $\boldsymbol{d}_k$ is given by

$$d_k[i] = \left(\frac{\delta g_p(\boldsymbol{s})}{\delta s[i]}\right)_{\boldsymbol{s}=\boldsymbol{s}_{k-1}} = p s_{k-1}[i]|s_{k-1}[i]|^{p-2} \quad (11)$$

where $s_{k-1}[i]$ is the ith element of $\boldsymbol{s}_{k-1}$. The Hessian is a diagonal matrix, and the ith element of the main diagonal of $\boldsymbol{H}_k$ is given by

$$H_k[i,i] = \left(\frac{\delta^2 g_p(\boldsymbol{s})}{\delta s[i]^2}\right)^{-1}_{\boldsymbol{s}=\boldsymbol{s}_{k-1}} = \frac{1}{p(p-1)}|s_{k-1}[i]|^{2-p}. \quad (12)$$

Note that $\boldsymbol{H}_k$ is defined for all values of $s_{k-1}[i]$ even when $s_{k-1}[i] = 0$. From (11) and (12) we find that $\boldsymbol{H}_k\boldsymbol{d}_k = \frac{1}{p-1}\boldsymbol{s}_{k-1}$. Substituting this expression into (10), we get

$$\begin{aligned} \boldsymbol{s}_k &= \frac{p}{p-1}\boldsymbol{s}_{k-1} - \frac{1}{p-1}\boldsymbol{H}_k\boldsymbol{A}^T(\boldsymbol{AH}_k\boldsymbol{A}^T)^{-1}\boldsymbol{As}_{k-1} \\ &= p_1\boldsymbol{s}_{k-1} + (1-p_1)\boldsymbol{H}_k\boldsymbol{A}^T(\boldsymbol{AH}_k\boldsymbol{A}^T)^{-1}\boldsymbol{x} \end{aligned} \quad (13)$$

where in the second equality $p_1 = \frac{p}{p-1}$, and we utilized the fact that $\boldsymbol{s}_{k-1}$ is feasible, i.e. $\boldsymbol{As}_{k-1} = \boldsymbol{x}$. The value $\boldsymbol{s}_k$ given by (13) is always feasible as shown by

$$\begin{aligned} \boldsymbol{As}_k &= p_1\boldsymbol{As}_{k-1} + (1-p_1)\boldsymbol{AH}_k\boldsymbol{A}^T(\boldsymbol{AH}_k\boldsymbol{A}^T)^{-1}\boldsymbol{x} \\ &= p_1\boldsymbol{x} + (1-p_1)\boldsymbol{x} = \boldsymbol{x}. \end{aligned}$$

By defining $\boldsymbol{W}_k = diag(|s_{k-1}[i]|^{1-0.5p})$, (13) can be written as

$$\boldsymbol{s}_k = p_1\boldsymbol{s}_{k-1} + (1-p_1)\boldsymbol{W}_k(\boldsymbol{AW}_k)^{\dagger}\boldsymbol{x}. \quad (14)$$

where † is the Moore-Penrose inverse.

Note that the resulting solution at the kth iteration is an affine combination of the previous solution and the generalized FOCUSS solution derived in [5] for the same objective function. Thus, MCCA is a regularized version of FOCUSS, where p_1 is the regularization parameter. Accordingly, instead of explicitly setting $p_1 = \frac{p}{p-1}$, we can choose to select the value of the regularization parameter such that $g_c(t)$ is minimized at every iteration. From the definitions of p_1 and $\boldsymbol{W}$, it may be shown that $-1 \le p_1 \le 0$ for *property 1* to be satisfied. The details of the proposed MCCA algorithm, along with the procedure to select p_1, are shown in Table 1.

Table I. MCCA Algorithm

Algorithm 1: MCCA Algorithm

Given an $(m \times n)$ matrix $\boldsymbol{A}$ of basis vectors, and a vector $\boldsymbol{x} \in \mathbb{R}^m$. Select a value for p such that $0 < p < 1$, a small threshold β, and an initial feasible point $\boldsymbol{s}_0$. This point can be selected as the least squares solution, i.e. $\boldsymbol{s}_0 = \boldsymbol{A}^T(\boldsymbol{AA}^T)^{-1}\boldsymbol{x}$. Then set $k = 0$ and repeat the following steps:

Start
1) Set $k = k + 1$,
2) Calculate $\boldsymbol{H}_k$ using (12).
3) Calculate $\tilde{\boldsymbol{s_k}} = \boldsymbol{H}_k\boldsymbol{A}^T(\boldsymbol{AH}_k\boldsymbol{A}^T)^{-1}\boldsymbol{x}$.
4) Calculate

$$\begin{aligned} p_1 &= \arg\min_q \quad g_p(q\boldsymbol{s}_k + (1-q)\tilde{\boldsymbol{s_k}}) \\ &\quad \text{subject to} \quad -1 \le q \le 0 \end{aligned}$$

5) Set $\boldsymbol{s}_k = p_1\boldsymbol{s}_k + (1-p_1)\tilde{\boldsymbol{s_k}}$.
6) if $||\boldsymbol{s}_k - \boldsymbol{s}_{k-1}||_2 < \beta$, break
else
go to step (1).
end
End

III-B. Performance Enhancement

The performance of (13) can be affected by the non–convexity of the objective function $g_p(\boldsymbol{s})$. As a result, MCCA may converge to a local minima. This problem can be alleviated by the following perturbation procedure. 1) select a feasible solution vector, 2) Run MCCA and get a new solution vector, 3) Perturb the solution vector by a random noise vector of suitable variance, that is constrained to be in the null space of $\boldsymbol{A}$ to ensure the feasibility of the new perturbed vector, 4) go to step 2 to obtain a new solution vector: 5) if the cardinality of the new solution vector is less than or equal to that of the previous solution vector, accept this new solution; otherwise, retain the old solution and go to step 3: 6) repeat steps 3–5 until a stopping criterion is satisfied. The stopping criterion could be a maximum number of iterations, or a pre–specified value of the cardinality of the solution.

As will be seen in the simulation results, following this procedure greatly improves the performance of the MCCA algorithm. When this perturbation procedure is followed, the resulting algorithm is called Iterative MCCA (IMCCA).

IV. SIMULATION RESULTS

To examine the effect of changing the value of p on the performance of MCCA and IMCCA algorithms, three

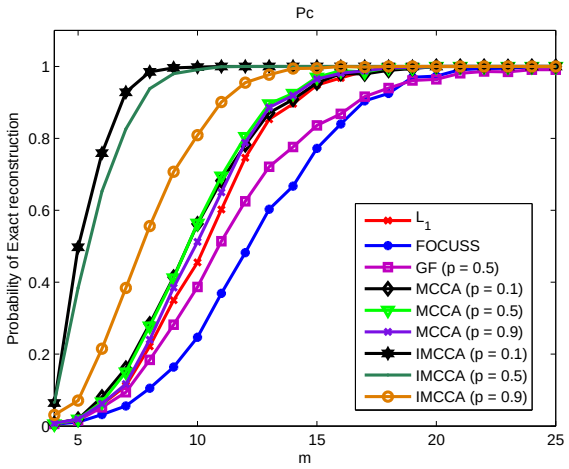

Fig. 2. Comparison between different algorithms in terms of the probability of exact reconstruction as a function of the number of observations.

different values of p are considered, e.g. $p = 0.1, 0.5$ and 0.9. The performance of the proposed algorithms is compared with those of the following three algorithms; 1) linear programming algorithm for minimizing ℓ_1-norm [6], 2) FOCUSS algorithm [3], and 3) generalized FOCUSS (GF) for minimizing $g_p(\boldsymbol{s})$ [5]. The cvx software[1] is utilized for solving (1) when $g(\boldsymbol{s})$ is the ℓ_1-norm.

The comparison between these three algorithms and the proposed algorithm is made in terms of the number of observations needed for exact reconstruction of the sparse vector. This example can be interpreted as finding the minimum number of sensors (m) needed for recovering n signals, where it is known that $n > m$ and the number of active sources at any time is $k < m$. In this example, $n = 40$, $k = 3$, and m increases from 4 to 25. For each value of m, a random $(m \times n)$ matrix $\boldsymbol{A}$ is created whose entries are each Gaussian random variables with zero mean and unit variance. A sparse solution $\boldsymbol{s}_s$ with k nonzero entries is then generated; the indices of these k entries are randomly selected, and their absolute amplitudes are chosen from a uniform distribution between 0.1 and 3, where their signs are randomly assigned. The same $\boldsymbol{A}$ and $\boldsymbol{s}_s$ are used for each algorithm. For each value of m, the probability of exact reconstruction, defined as the ratio between the number of runs at which each algorithm estimated $\boldsymbol{s}_s$ successfully to the total number of runs, is calculated. The total number of runs in this example is 1000.

The results are shown in Figure 2, where in this figure, "GF" refers to the generalized FOCUSS algorithm [5]. As shown in this figure, although the performance of MCCA is almost similar to that of ℓ_1-norm, IMCCA algorithm, for all considered values of p, outperform the other algorithms. Moreover, the smaller the considered value of p, the better the performance of IMCCA. Moreover, the result that IMCCA significantly outperforms MCCA indicates that convergence to local minima is a significant problem and that the perturbation technique offered by IMCCA is at least partially effective in dealing with this issue.

[1]This is a free software available at http://www.stanford.edu/˜boyd/cvx/

It is shown in Figure 2 that the minimum value of m for which the IMCCA algorithm with $p = 0.1$ (best case) is capable of correctly estimating the sparse vector $\boldsymbol{s}_s$ is 8 sensors, while both FOCUSS and the GF algorithms each require 22 sensors. The ℓ_1 algorithm requires 18 sensors.

V. CONCLUSION

In this paper we presented a new technique for minimizing a class of nonconvex functions for solving the problem of under–determined system of linear equations. The proposed technique was based on locally replacing the nonconvex objective function by a convex objective function. The resulting algorithm is iterative and outperforms some previously-developed algorithms that are utilized for solving the same problem.

VI. REFERENCES

[1] S. Chen and D. Donoho, "Basis pursuit," in Proc. Twenty-Eighth Asilomar Conf. Signals, Syst. Comput., Monterey, CA, Nov. 1994, vol. I, pp. 41-44.

[2] P. Xu and D. Yao, "Development and evaluation of the sparse decomposition method with mixed over-complete dictionary for evoked potential estimation," Computers in Biology and Medicine, vol. 37, pp. 1731-1740, 2007.

[3] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstructions from limited data using FOCUSS: A re-weighted minimum norm algorithm," IEEE Trans. Signal Process., vol. 45, pp. 600-616, Mar. 1997.

[4] Y. Li, A. Cichocki, and S. Amari, "Analysis of sparse representation and blind source separation," Neural Computation, vol. 16, pp. 1193-1234, 2004.

[5] B. D. Rao and K. Kreutz-Delgado, "An affine scaling methodology for best basis selection," IEEE Trans. Signal Process., vol. 47, no. 1, Jan. 1999.

[6] S. Boyd and L. Vandenberghe, "Convex Optimization", Cambridge University Press. Available online at www.stanford.edu/ boyd/cvxbook/bv_cvxbook.pdf.

[7] D. L. Donoho, "For most underdetermined systems of linear equations, the minimal ℓ^1 solution is also the sparsest solution." Manuscript. Submitted. URL: http://wwwstat. stanford.edu/donoho/Reports/2004

[8] R. Chartrand, "Exact reconstruction of sparse signals via nonconvex minimization," IEEE Signal Process. Lett., vol. 14, no. 10, Oct. 2007.