

Lagrange Programming Neural Network for the l_1 -norm Constrained Quadratic Minimization

Ching Man Lee, Ruibin Feng, and Chi-Sing Leung^(✉)

Department of Electronic Engineering, City University of Hong Kong,
Kowloon Tong, Hong Kong

rfeng4-c@my.cityu.edu.hk, eeleungc@cityu.edu.hk

Abstract. The Lagrange programming neural network (LPNN) is a framework for solving constrained nonlinear programming problems. But it can solve differentiable objective/constraint functions only. As the l_1 -norm constrained quadratic minimization (LICQM), one of the sparse approximation problems, contains the nondifferentiable constraint, the LPNN cannot be used for solving LICQM. This paper formulates a new LPNN model, based on introducing hidden states, for solving the LICQM problem. Besides, we discuss the stability properties of the new LPNN model. Simulation shows that the performance of the LPNN is similar to that of the conventional numerical method.

Keywords: Stability · Sparse approximation · LPNN

1 Introduction

Throughout the years, there are many studies on solving constrained nonlinear programming problems [1–3] based on neural circuit approach. When the real-time solutions are needed, the analog neural circuit is preferred. Although many neural models [4, 5] were introduced, they are designed for a particular kind of problems only. The Lagrange programming neural network (LPNN) [3, 6, 7] is a framework for solving constrained nonlinear programming problems. But it can solve differentiable objective/constraint functions only.

In signal processing, one of the important topics is sparse approximation [8, 9]. Sparse approximation aims at recovering an unknown sparse signal from the measurements. Sparse approximation has many potential applications, such as channel estimation in MIMO wireless communication channels [10] and image denoising [11]. However, in sparse approximation, many problems involve nondifferentiable objective or constraints. By introducing the concept of soft threshold function, the local competition algorithm (LCA) [12], being an analog method, is able to solve the basis pursuit denoising problem [9], which is an unconstrained nonlinear programming problem. However, the LCA is not designed to handle constrained optimization problems.

This paper focuses on l_1 -norm constrained quadratic minimization (L1CQM) for sparse approximation. Section 2 reviews the basic concept of LPNN, sparse approximation, and LCA. Section 3 presents the proposed LPNN model for solving the L1CQM. Section 4 presents some properties of the LPNN. Simulation results are then presented in Sect. 5.

2 Background

LPNN: The LPNN approach considers a general constrained nonlinear programming problem:

$$\text{EP: } \min f(\mathbf{x}) \quad \text{s.t. } \mathbf{h}(\mathbf{x}) = \mathbf{0}, \quad (1)$$

where $\mathbf{x} \in \mathfrak{R}^n$ is the state vector, $f : \mathfrak{R}^n \rightarrow \mathfrak{R}$ is the objective function, and $\mathbf{h} : \mathfrak{R}^n \rightarrow \mathfrak{R}^m$ ($m < n$) describes the m equality constraints. The objective function f and constraints \mathbf{h} are assumed to be twice differentiable. It should be noticed that the LPNN approach can solve inequality constraints by introducing dummy variables.

In LPNN, a Lagrangian function is set up, given by

$$\mathcal{L}_{ep} = f(\mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{h}(\mathbf{x}), \quad (2)$$

where $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_m]^T$ is the Lagrange multiplier vector. There are two kinds of neurons: variable neurons and Lagrange neurons. The variable neurons hold the variable vector \mathbf{x} , while the Lagrange neurons hold the multiplier vector $\boldsymbol{\lambda}$. The LPNN dynamics are given by

$$\frac{1}{\epsilon} \frac{d\mathbf{x}}{dt} = -\frac{\partial \mathcal{L}_{ep}}{\partial \mathbf{x}}, \quad \text{and} \quad \frac{1}{\epsilon} \frac{d\boldsymbol{\lambda}}{dt} = \frac{\partial \mathcal{L}_{ep}}{\partial \boldsymbol{\lambda}}, \quad (3)$$

where ϵ is the time constant of the circuit. In this paper, ϵ is considered as equal to 1 in regard to generality.

Sparse Approximation: In sparse approximation, we would like to estimate a sparse solution $\mathbf{x} \in \mathfrak{R}^n$ from the measurement $\mathbf{b} = \boldsymbol{\Phi} \mathbf{x} + \boldsymbol{\xi}$, where $\mathbf{b} \in \mathfrak{R}^m$ is the observation vector, $\boldsymbol{\Phi} \in \mathfrak{R}^{m \times n}$ is the measurement matrix with a rank of m , $\mathbf{x} \in \mathfrak{R}^n$ is the unknown sparse vector ($m < n$), and ξ_i 's are the measurement noise. The recovery can be formulated as the following programming problem:

$$\min \|\mathbf{b} - \boldsymbol{\Phi} \mathbf{x}\|_2^2, \quad \text{s.t. } \|\mathbf{x}\|_1 \leq \psi, \quad (4)$$

where $\psi > 0$. This problem is called as l_1 -norm Constrained Quadratic Minimization (L1CQM). In this problem, we would like to minimize the residue subject to the sum of absolute of signal elements is less than a value. Since the constraint function $\|\mathbf{x}\|_1 - \psi$ is nondifferentiable, the conventional LPNN is unable to solve L1CQM directly.

Subdifferential: Subdifferential was developed to handle nondifferentiable functions. Their definitions are stated in the following.

Definition 1. Given a convex function f , the subgradient $\boldsymbol{\rho}$ of f at \mathbf{x} is

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \boldsymbol{\rho}^T(\mathbf{y} - \mathbf{x}), \forall \mathbf{y}. \quad (5)$$

Definition 2. The subdifferential $\partial f(\mathbf{x})$ at \mathbf{x} is the set of all subgradients:

$$\partial f(\mathbf{x}) = \{ \boldsymbol{\rho} \mid \boldsymbol{\rho}^T(\mathbf{y} - \mathbf{x}) \leq f(\mathbf{y}) - f(\mathbf{x}), \forall \mathbf{y} \}. \quad (6)$$

Note that when $f(\cdot)$ is differentiable at \mathbf{x}_o , its subdifferential at \mathbf{x}_o is equal to the conventional partial derivative. Let us use the absolute function $f(x) = |x|$ as an example to example the subdifferential:

$$\partial|x| = \begin{cases} [-1, 1] & x = 0, \\ \text{sign}(x) & x \neq 0. \end{cases} \quad (7)$$

Concept of LCA: The LCA is designed to handle the following unconstrained optimization problems:

$$\mathcal{L} = \frac{1}{2} \|\boldsymbol{\Phi}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{x}\|_1, \quad (8)$$

where λ is a trade-off parameter. In the LCA, there are n neurons. The neuron outputs are denoted as \mathbf{x} and their internal states are denoted as \mathbf{u} . By a threshold function, the mapping from \mathbf{u} to \mathbf{x} is stated as below

$$x_i = T_\lambda(u_i) = \begin{cases} 0, & \text{for } |u_i| \leq \lambda, \\ u_i - \lambda \text{sign}(u_i), & \text{for } |u_i| > \lambda. \end{cases} \quad (9)$$

The forward mapping T_λ from u_i to x_i is one-to-one for $|u_i| > \lambda$ and is many-to-one for $|u_i| \leq \lambda$. The inverse mapping T_λ^{-1} from x_i to u_i is one-to-one for $x_i \neq 0$, while it is one-to-many for $x_i = 0$. It implies that $T_\lambda^{-1}(0)$ is equal to a set: $[-\lambda, \lambda]$. The LCA defines the dynamics on \mathbf{u} rather than \mathbf{x} , given by

$$\frac{d\mathbf{u}}{dt} = -\partial_{\mathbf{x}}\mathcal{L}_{lca} = -\lambda\partial|\mathbf{x}|_1 + \boldsymbol{\Phi}^T(\mathbf{b} - \boldsymbol{\Phi}\mathbf{x}). \quad (10)$$

With the property of $T_\lambda(\cdot)$ and the definition subdifferential [12], the LCA replaces “ $\kappa\partial|\mathbf{x}|_1$ ” with “ $\mathbf{u} - \mathbf{x}$ ”. The dynamics become

$$\frac{d\mathbf{u}}{dt} = -\mathbf{u} + \mathbf{x} + \boldsymbol{\Phi}^T(\mathbf{b} - \boldsymbol{\Phi}\mathbf{x}). \quad (11)$$

If we do not introduce the internal state vector \mathbf{u} , $\partial|\mathbf{x}|_1$ (may be equal to a set) is not implementable.

3 LPNN for L1CQM

Our aim is to solve the programming problem:

$$\min \|\mathbf{b} - \boldsymbol{\Phi}\mathbf{x}\|_2^2, \text{ s.t. } \|\mathbf{x}\|_1 \leq \psi. \quad (12)$$

From the convex optimization theory, one can obtain the following theorem.

Theorem 1. For the programming problem (12), \mathbf{x}^* is an optimal solution, iff, there exists a λ^* (Lagrange multiplier) and

$$\mathbf{0} \in -2\Phi^T(\mathbf{b} - \Phi\mathbf{x}^*) + \lambda^*(\partial|\mathbf{x}^*|_1), \quad (13a)$$

$$|\mathbf{x}^*|_1 - \psi \leq 0, \quad (13b)$$

$$\lambda^* \geq 0, \quad (13c)$$

$$\lambda^*(|\mathbf{x}^*|_1 - \psi) = 0. \quad (13d)$$

where λ^* is the optimal dual variable (Lagrange multiplier).

Since the problem (12) has an inequality constraint, we cannot directly use the LPNN. However, if ψ^2 is less than a certain value, given by $\psi^2 < \frac{|\Phi^T \mathbf{b}|_2^2}{|\Phi^T \Phi|_2^2}$, then the inequality constraint becomes an equality ones. Therefore, Theorem 1 becomes the following theorem.

Theorem 2. If $\psi^2 < \frac{|\Phi^T \mathbf{b}|_2^2}{|\Phi^T \Phi|_2^2}$, then the optimization problem (13) becomes

$$\min |\mathbf{b} - \Phi\mathbf{x}|_2^2, \quad s.t. |\mathbf{x}|_1 = \psi. \quad (14)$$

And \mathbf{x}^* is the optimal solution, iff, there exists a λ^* (Lagrange multiplier) and

$$\mathbf{0} \in -2\Phi^T(\mathbf{b} - \Phi\mathbf{x}^*) + \lambda^*(\partial|\mathbf{x}^*|_1), \quad (15a)$$

$$|\mathbf{x}^*|_1 - \psi = 0, \quad (15b)$$

$$\lambda^* > 0. \quad (15c)$$

Note that (15) summarizes the KKT conditions (necessary and sufficient).

Proof: Suppose $(\mathbf{x}^*, \lambda^*)$ is an optimal solution of Theorem 1. That means, $\lambda^* \geq 0$. Firstly, we will use contradiction to prove that λ^* cannot be equal to zero when $\psi^2 < \frac{|\Phi^T \mathbf{b}|_2^2}{|\Phi^T \Phi|_2^2}$.

Since $(\mathbf{x}^*, \lambda^*)$ are optimal, they satisfy the KKT conditions (14) of Theorem 1. If $\lambda^* = 0$, then from (13a) we have

$$\Phi^T \Phi \mathbf{x}^* = \Phi^T \mathbf{b} \Rightarrow |\Phi^T \Phi|_2 |\mathbf{x}^*|_2 \geq |\Phi^T \mathbf{b}|_2 \Rightarrow |\mathbf{x}^*|_2^2 \geq \frac{|\Phi^T \mathbf{b}|_2^2}{|\Phi^T \Phi|_2^2}. \quad (16)$$

From (13b),

$$\psi^2 \geq |\mathbf{x}^*|_1^2 = \left(\sum_{i=1}^n |x_i^*| \right)^2 = \sum_{i=1}^n x_i^{*2} + \sum_{i=1}^n \sum_{i \neq j}^n |x_i^*| |x_j^*| \geq \sum_{i=1}^n x_i^{*2} \geq |\mathbf{x}^*|_2^2 \quad (17)$$

From (16) and (17),

$$\psi^2 \geq |\mathbf{x}^*|_2^2 \Rightarrow \psi^2 \geq \frac{|\Phi^T \mathbf{b}|_2^2}{|\Phi^T \Phi|_2^2}. \quad (18)$$

The above contradicts the assumption of $\psi^2 < \frac{|\Phi^T \mathbf{b}|_2^2}{|\Phi^T \Phi|_2^2}$. As a result, it proves that $\lambda^* > 0$. This means, the KKT conditions of Theorem 1 can be rewritten as (15). Besides, the inequality in (12) can be removed. Thus, the optimization problem can be written as (14). The proof is complete. ■

With Theorem 2, we can define the Lagrangian function:

$$\mathcal{L}_n = |\mathbf{b} - \Phi \mathbf{x}|_2^2 + \lambda(|\mathbf{x}|_1 - \psi). \quad (19)$$

Since the traditional LPNN model cannot handle nondifferentiable constraints, direct implementing the neuron dynamics is impossible. Using the concept of the LCA, we introduce the hidden state vector \mathbf{u} as the internal state vector and \mathbf{x} as the corresponding neuron outputs. Besides, from the property of $T_\kappa(\cdot)$ and the definition subdifferential [12], the dynamics of \mathbf{u} and λ is given by

$$\frac{d\mathbf{u}}{dt} = 2\Phi^T(\mathbf{b} - \Phi \mathbf{x}) - \lambda(\mathbf{u} - \mathbf{x}) \quad (20a)$$

$$\frac{d\lambda}{dt} = |\mathbf{x}|_1 - \psi. \quad (20b)$$

The role of (20a) is used to minimize the objective value, while the role of (20b) is used to constraint \mathbf{x} in the feasible region.

We can use Fig. 1 to illustrate the idea of LPNN for L1CQM. In Fig. 1(a), there is a 1D sparse signal. The length of this signal is 128. There are five nonzero elements. The number of measurement is 30. The measurement matrix is an ± 1 random matrix. In Fig. 1(b), we show the recovery from the pseudoinverse. Clearly it is not our expected signal. In Fig. 1(c), we show the recovery signal

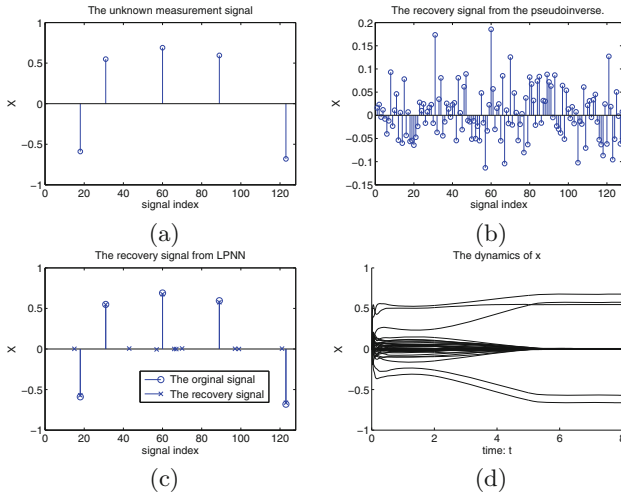


Fig. 1. Illustration example for the LPNN. (a) The original signal. (b) Recovery from pseudoinverse. (c) Recovery from LPNN. (d) The dynamics of LPNN.

from the LPNN, which is pretty close to the original signal. Figure 1(d) shows the dynamics of \mathbf{x} . It can be seen that the LPNN can settle down in five characteristic time.

4 Properties of LPNN

This section discusses the properties of the proposed LPNN for L1CQM. Firstly, we will show that the equilibrium point of the LPNN approach is the global minimum point of the L1CQM.

Theorem 3. *Let $(\mathbf{u}^*, \lambda^*)$, with $\lambda^* > 0$ and $\mathbf{u}^* \neq \mathbf{0}$, be an equilibrium point of the LPNN dynamics (Eq. (20a)) and \mathbf{x}^* be the corresponding output vector. The KKT conditions in Theorem 2 are satisfied at this equilibrium point. And the corresponding output vector \mathbf{x}^* is the optimal solution of the problem.*

Proof: In the proof, we will prove that $(\mathbf{u}^*, \lambda^*)$, with $\lambda^* > 0$ and $\mathbf{u}^* \neq \mathbf{0}$ satisfies the KKT conditions in Theorem 2. At the equilibrium point, from (20b), we have

$$2\Phi^T(\mathbf{b} - \Phi\mathbf{x}^*) - \lambda^*(\mathbf{u}^* - \mathbf{x}^*) = \mathbf{0}, \quad (21)$$

$$|\mathbf{x}^*|_1 - \psi = 0. \quad (22)$$

For (21) and by $\partial|\mathbf{x}|_1 = \mathbf{u} - \mathbf{x}$,

$$2\Phi^T(\mathbf{b} - \Phi\mathbf{x}^*) - \lambda^*(\mathbf{u}^* - \mathbf{x}^*) = \mathbf{0} \Rightarrow -2\Phi^T(\mathbf{b} - \Phi\mathbf{x}^*) + \lambda^*(\partial|\mathbf{x}^*|_1) = \mathbf{0}.$$

The above means that “ $\mathbf{0} \in -2\Phi^T(\mathbf{b} - \Phi\mathbf{x}^*) + \lambda^*(\partial|\mathbf{x}^*|_1)$ ”. This satisfies the KKT condition (15a). Based on the assumption that $\lambda^* > 0$, (15c) is satisfied. Also for “ $|\mathbf{x}^*|_1 - \psi = 0$ ” in (22), it satisfies the KKT condition (15b). As the KKT conditions are necessary and sufficient, which implies that any equilibrium point $(\mathbf{u}^*, \lambda^*)$, with $\mathbf{u}^* \neq \mathbf{0}$ and $\lambda^* > 0$, is an optimal solution to the problem. The proof is complete. \blacksquare

Another thing needed to be concern is the stability of the equilibrium point. Otherwise, the equilibrium points of the LPNN are not achievable. Based on the approach in [3, 6], the equilibrium point of (20) is an asymptotically stable point.

5 Simulations

The proposed LPNN approach is undergoing some experiments by using the standard configs [13, 14]. The aim of the experiment is to verify if the proposed LPNN has the similar performance to the conventional numerical method LASSO. For the sparse vector $\mathbf{x} \in \mathbb{R}^n$, we consider two signal lengths: $n = 512, 4096$. The numbers of non-zero elements in \mathbf{x} are selected as 15 and 25 for $n = 512$, and 75 and 125 for $n = 4096$, respectively. For the non-zero elements, they are uniformly distributed random numbers, either in between -5 to -1 or in between 1 to 5 .

In the tests, the measurement matrix is an ± 1 random matrix, which is then normalized with the signal length. In the experiments, we vary the number m of measurement signals. For each setting, we repeat the experiments with 100 times using different measurement matrices and sparse signals. The variances of Gaussian noise introduced in the measured signal are $\sigma^2 = \{0.05^2, 0.025^2, 0.005^2\}$.

In order to compare the proposed LPNN approach with digital method, the LASSO algorithm from SPGL1 [13, 14] is applied for recovering the sparse signals. During comparison, the mean square error (MSE) values of recovery signals are recorded. The results are shown in Figs. 2 and 3. From the figures, the performance of the LPNN is similar to that of the LASSO algorithm. In addition, as shown in Fig. 2, for $n = 512$ with 15 non-zero data points, around 80 measurements are required to recover the sparse signal. For 25 non-zero data points, around 120 measurements are required. As shown in Fig. 3, for $n = 4096$ with 75 non-zero data points, around 500 measurements are required to recover the sparse signal. For 125 non-zero data points, around 700 measurements are required.

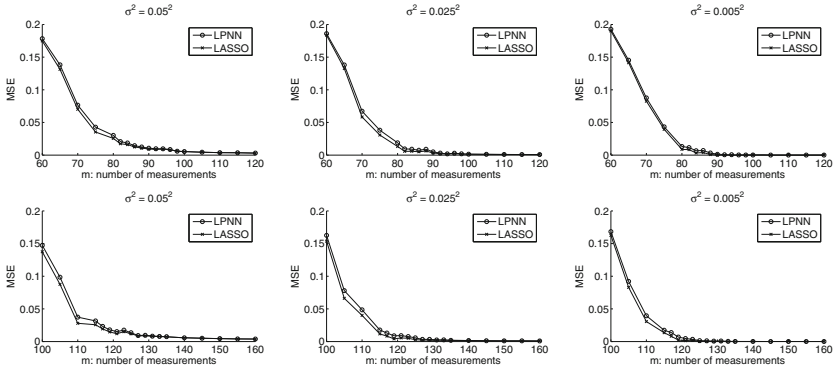


Fig. 2. The MSE of the recovery signals from noisy measurement for signal length $n = 512$. The signals contain 15 and 25 non-zero data points in first and second row.

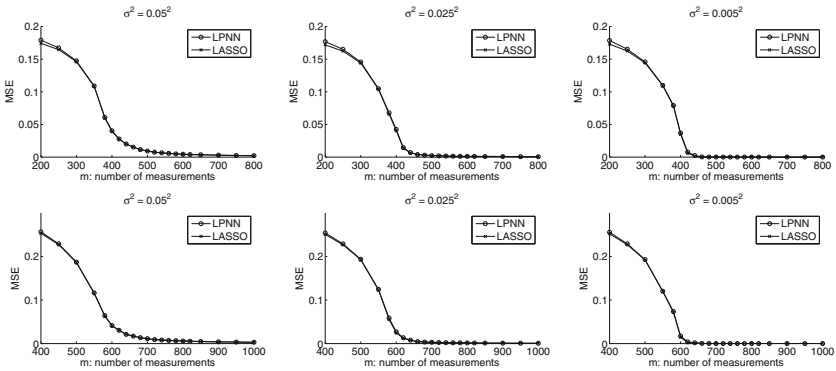


Fig. 3. The MSE of the recovery signals from noisy measurement for signal length $n = 4096$. The signals contain 75 and 125 non-zero data points in first and second row.

6 Conclusion

This paper proposed a new LPNN model for solving the L1CQM problem. In the theoretical side, we proved that the equilibrium points of the LPNN is the optimal solution. Besides, stimulations are carried out to verify the effectiveness of the LPNN. There are some possible extensions of our works. From the simulation, the network always converges. However, we do not theoretically show that the LPNN is global stable. Hence it is interesting to theoretically study the global stability.

Acknowledgement. The work was supported by a research grant from City University of Hong Kong (Project No.: 7004233).

References

1. Chua, L.O., Lin, G.N.: Nonlinear programming without computation. *IEEE Trans. Circuits Syst.* **31**, 182–188 (1984)
2. Xiao, Y., Liu, Y., Leung, C.S., Sum, J., Ho, K.: Analysis on the convergence time of dual neural network-based kwta. *IEEE Trans. Neural Netw. Learn. Syst.* **23**(4), 676–682 (2012)
3. Leung, C.S., Sum, J., Constantinides, A.G.: Recurrent networks for compressive sampling. *Neurocomputing* **129**, 298–305 (2014)
4. Gao, X.B.: Exponential stability of globally projected dynamics systems. *IEEE Trans. Neural Netw.* **14**, 426–431 (2003)
5. Hu, X., Wang, J.: A recurrent neural network for solving a class of general variational inequalities. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **37**(3), 528–539 (2007)
6. Zhang, S., Constantinides, A.G.: Lagrange programming neural networks. *IEEE Trans. Circuits Syst. II* **39**(7), 441–452 (1992)
7. Leung, C.S., Sum, J., So, H.C., Constantinides, A.G., Chan, F.K.W.: Lagrange programming neural network approach for time-of-arrival based source localization. *Neural Comput. Appl.* **12**, 109–116 (2014)
8. Donoho, D.L., Elad, M.: Optimally sparse representation in general (nonorthogonal) dictionaries via l_1 minimization. *Proc. Nat. Acad. Sci.* **100**(5), 2197–2202 (2003)
9. Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.* **20**(1), 33–61 (1998)
10. Gilbert, A.C., Tropp, J.A.: Applications of sparse approximation in communications. In: *Proceedings International Symposium on Information Theory ISIT 2005*, pp. 1000–1004 (2005)
11. Rahmoune, A., Vandergheynst, P., Frossard, P.: Sparse approximation using m-term pursuit and application in image and video coding. *IEEE Trans. Image Process.* **21**, 1950–1962 (2012)
12. Rozell, C.J., Johnson, D.H., Baraniuk, R.G., Olshausen, B.A.: Sparse coding via thresholding and local competition in neural circuits. *Neural Comput.* **20**(10), 2526–2563 (2008)
13. Ji, S., Xue, Y., Carin, L.: Bayesian compressive sensing. *IEEE Trans. Signal Process.* **56**(6), 2346–2356 (2007)
14. van den Berg, E., Friedlander, M.P.: Probing the Pareto frontier for basis pursuit solutions. *SIAM J. Sci. Comput.* **31**(2), 890–912 (2008)