ATLANTIS PRESS
Advances in Intelligent Systems Research, volume 143
2nd International Conference on Applied Mathematics, Modelling and Statistics Application (AMMSA 2018)

Compressive Sensing Method for Function Recovery

Aitong Huang and Renzhong Feng
School of Mathematics and Systematic Science, Beihang University, Beijing 100083, China

***Abstract*—It is well known that under certain orthogonal systems (such as Chebyshev tensor and Legendre polynomial space), the expansion coefficient of a smooth function has a sparseness that the coefficient with a finite number of coefficients after the first is gradually zero. For accurate sampling and sampling data with noise, this paper uses compressive sensing technology to recover the first limited number of function expansion coefficients, so as to achieve the purpose of function recovery. Numerical experiments show that this technique is feasible.**

Keywords—function recovery; compressed sensing; l_1 minimization problem; orthogonal matching pursuit algorithm

I. INTRODUCTION

The problem of function recovery has always been a hot topic in the area of approximation theory. For the problem of function recovery, there are many mature methods to solve them, such as the most commonly used methods of interpolation, quasi-interpolation, and least squares, moving least squares, and they all have good properties. For example, the interpolation has the property of accurately recovering the interpolation node, and the least squares have the best approximation in the norm meaning. However, when the number of sampling points of the function to be recovered is much smaller than the number of basis functions in the base function space, the above method cannot be fully applied during function recovery. Therefore, consider the problem of shifting the model of the least square problem to the l_1 norm, as in [1]. At the same time, with the gradual maturity and development of compressed sensing technology, the use of low-dimensional vectors to recover vectors in high dimensions has gradually become a reality. It is well-known that high-dimensional multivariate functions have sparse unfolding properties in certain orthogonal systems (eg Chebyshev tensor and Legendre polynomial space),as in [2]. Therefore, in recent years, the use of compressed sensing technology to accurately approximate functions has gradually become a hot topic of discussion. The standard compressive sensing problem itself is calculated in a finite space that is predicted in advance, but the function is in an infinite-dimensional space. Therefore, there are certain problems in directly using the standard compressive sensing problem to solve the problem of approximation. At the same time, it was found that under a particular orthogonal system, functions in an infinite-dimensional space can be approximated by the sum of their finite terms, as in [2]. Through this feature, we transform the original problem in infinite-dimensional space into a finite-dimensional space, and then use the existing compressive sensing technology to solve it.

II. FUNCTION RECOVERY PROBLEM MODEL

Let function $f(t)$ be a continuous function whose domain is D. Then according to Fourier series expansion, the function can be written as $f(t)=\sum_{j=1}^{\infty} c_j \varphi_j(t)$, where $\{\varphi_j(t)\}_{j=1}^{\infty}$ is a set of orthogonal basis functions. After observing $c=\{c_j\}_{j=1}^{\infty}$, we found that there are only a limited number of coefficients $c_i = O(1)$, most of the coefficients tend to 0, so denote the index set as $I=\mathbb{N}, I_n=\{i, c_i = O(1)\}$. If we can design an approximation method that can give $\tilde{c}_i \approx c_i, i \in I_n$, then $\tilde{f}(t)=\sum_{j\in I_n} \tilde{c}_j \varphi_j(t)$ is a good approximation for $f(t)$.

First, consider the case where there is no noise at the sampling point. At this time, the sampling value is completely accurate. The model for minimizing the l_1 function recovery is:

$$\begin{aligned} &\min \|c\|_1 \\ &s.t.\ \ Ac=b \end{aligned} \tag{1}$$

Here

$$A=\begin{bmatrix} \varphi_1(t_1) & \cdots & \varphi_n(t_m) \\ \vdots & \ddots & \vdots \\ \varphi_1(t_m) & \cdots & \varphi_n(t_m) \end{bmatrix}, b=\begin{bmatrix} f(t_1) \\ \vdots \\ f(t_m) \end{bmatrix}$$

$t=[t_1 \ \cdots \ t_m]^T$ is to define a set of sampling points within domain D, the vector $c=[c_1 \ \cdots \ c_n]^T$ consisting of the coefficients of the basis functions.

However, in actual sampling, the sampling value is usually a sampling value containing noise. Therefore, the above problem model can be converted into

$$\begin{aligned} &\min \|c\|_1 \\ &s.t.\ \ \|Ax-b\|_2 \le \delta \end{aligned} \tag{2}$$

Here, the definition of A, c is consistent with the definition in the previous article, and b is the sample value with error, that is,

$$b = \left[f(t_1)+e_1 \quad \cdots \quad f(t_m)+e_m \right]^T$$

The noise vector is $e = \left[e_1 \quad \cdots \quad e_1 \right]^T$, δ is the upper bound of the error 2-norm given by the sample value in advance.

III. Compressed Sensing Problem Model and Solution Method

Given a coding and decoding (Δ, A) , we care about its performance, ie

$$\left\| x_0 - \Delta(Ax_0) \right\|_X \tag{3}$$

Here X is a given norm. When the number of non-zero elements in x0 is small, a more natural decoding $\Delta_0(b)$ is the solution to the following linear programming problem:

$$\begin{aligned} P_0:\ \ &\min_{x\in R^N} \|x\|_0 \\ &s.t.\ \ Ax = b \end{aligned} \tag{4}$$

Here, $\|x\|_0$ represents the number of non-zero elements in x. But problem P_0 is an NP complete problem. So, can we find a more effective decoding method? One surprising fact is that if matrix A meets certain conditions, the answer is yes, but we have to pay a price for the number of observations. We now define decoding $\Delta_1(b)$ as a solution to the following problem:

$$\begin{aligned} P_1\ \ &\min_{x\in R^N} \|x\|_1 \\ &s.t.\ \ Ax = b \end{aligned} \tag{5}$$

A natural question is: Is the solution of P_1 equivalent to the solution of P_0? Or, for what kind of observation matrix A, P_1 solution and P_0 solution always agree?

Before answering the above question, we first give the definition of the matrix RIP properties, as in [3]: we say that matrix A satisfies s-order RIP properties, if there are constants $\delta_s \in [0,1)$, satisfies

$$(1-\delta_s)\|x\|_2^2 \le \|Ax\|_2^2 \le (1+\delta_s)\|x\|_2^2$$

A sufficient condition for accurately recovering the s-factor signal is given below, as in [3]: Assume that the coding matrix A satisfies the 2s-order RIP property, and $\delta_{2s} \le \sqrt{2}-1$, we choose to decode Δ_1 . Well, for any $x \in \sum_s := \{x \in R^N : \|x\|_0 \le s\}$, we have

$$\Delta_1(Ax) = x$$

With these safeguards in mind, we consider using the method of solving the compressive sensing problem to solve the problem of l1 minimization. Some iterative methods for quickly solving compressive sensing problems, such as Bregman iterative algorithms and ADM (alternating direction method) algorithms, have been matured, as in [4][5]. In this paper, we mainly use another decoding algorithm: greedy algorithm. Orthogonal matching algorithm is a commonly used greedy algorithm, which usually solves the approximate solution of the following problem:

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} \|x\|_0 \\ &s.t.\ \ \|Ax-b\|_2^2 \le \delta \end{aligned} \tag{6}$$

Here $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$ are given matrices and vectors. The basic idea is to select the least number of columns from A so that they form an approximate representation of the pair. The method first computes the orthogonal projection of b in the selected column expansion space and then computes the orthogonal projection complement and A. The magnitude of the absolute value of the product within the column, usually selected to maximize the absolute value of the inner product. The algorithm process is as follows, as in [3]:

Algorithm 1: OMP

Input: encoding matrix A, vector b, maximum sparsity s

Output: Restored Vector x^*

Initial value: $r^0 = b, c^0 = 0, \Lambda^0 = \varnothing, l = 0$

$while\ l \le s$

$match: h = A^T r^l$

$identity: \Lambda^{l+1} = \Lambda^l \cup \left\{ \arg\max_j \left| h^l(j) \right| \right\}$

$update: c^{l+1} = \arg\min_{z:\sup p(z)\subset\Lambda^{l+1}} \|b - Az\|$

$r^{l+1} = b - Ac^{l+1}$

$l = l+1$

$end\ while$

$x^* = c^{s+1}$

The convergence of the above algorithm has been guaranteed by the theorem, as in [5].

Theorem : Assume that $0<\delta<1$, and A satisfies the RIP condition $\delta_{2s}+(1+\delta)\delta_{2\alpha s}\le\delta$. So for any $x\in\mathbb{R}^n$, there are

$$\left\|OMP_{2(\alpha-1)s}(Ax)-x\right\|_2\le\frac{C_2\sigma_s(x)_1}{\sqrt{s}} \tag{7}$$

Here $\alpha=\lceil 16+15\delta\rceil$, $\sigma_s(x)_1=\min_{z\in\sum_s}\|x-z\|_1$, $\sum_s:=\{x\in\mathbb{R}^n:\|x\|_0\le s\}$ and $C_2=2(1+\delta)\left((1+20\delta)^{1/2}+1\right)+3$.

IV. Numerical Experiments

This section mainly conducts numerical experiments on two types of functions. The first type is the function in the finite Wiki function space for the restored function; the second type is any given smooth function for the recovered function. In a numerical experiment, the Orthogonal Matching Pursuit (OMP) algorithm of Decoder's Δ_1 is used to obtain an approximate vector of the vector c, and an approximate function of the function to be recovered is obtained. And in order to verify the validity of the method, the errors used in the experiments in this section are:

(a) $L_\infty-error1:=\max_{1\le j\le m}\left|f(t_j)-\tilde{f}(t_j)\right|, L_2-error1:=\left\|f(t)-\tilde{f}(t)\right\|_2$, here $f(t)=\left[f(t_1),\cdots,f(t_m)\right]^T$, $\tilde{f}(t)=\left[\tilde{f}(t_1),\cdots,\tilde{f}(t_m)\right]^T$;

(2) $L_\infty-error2:=\max_{1\le i\le n}|c-\tilde{c}|$, $L_2-error2:=\|c-\tilde{c}\|_2/\|c\|_2$,here $c=\left[c_1\cdots,c_n\right]^T$, $\tilde{c}=\left[\tilde{c}_1,\cdots,\tilde{c}_n\right]^T$.

The two orthogonal basis functions that we use in numerical experiments are: Legendre basis function and Chebyshev basis function. Their recurrence formulas are::

(1) Legendre basis function:

$$\begin{aligned}&P_0(t)=1\\&P_1(t)=t\\&(n+1)P_{n+1}(t)=(2n+1)tP_n(t)-nP_{n-1}(t),n=1,2,\cdots\end{aligned} \tag{8}$$

(2) Chebyshev basis function:

$$\begin{aligned}&T_0(t)=1\\&T_1(t)=\cos(\arccos t)=t\\&T_{n+1}(t)=2tT_n(t)-T_{n-1}(t),\quad n=1,2,\cdots\end{aligned} \tag{9}$$

After the above two sets of basis functions have been defined, an estimation of the number of sampling points is also needed. In order to successfully use the decoder to recover the coefficients of the function, according to the literature [compressed sensing], the following relationship must be satisfied between the number of sampling points m and the number of basis functions N and vector sparsity s:

$$m\ge O\left(s\log\left(\frac{N}{s}\right)\right) \tag{10}$$

Next, numerical experiments are performed for the above two types of to-be-restored functions in the cases where the sampled value is noisy and the sampled value is no-noise, and the validity of the method is verified.

Experiment 1: The sampled value does not contain noise: In the numerical experiment, n=512, s=14, and m=128. That is, the selection of Legendre basis function is $\{P_j(t)\}_{j=1}^{512}$,the Chebyshev basis function is $\{T_j(t)\}_{j=1}^{512}$. In the interval [-1, 1], 128 points $\{t_i\}_{i=1}^{128}$ with uniform distribution are selected, and the sparsity of vector c is selected. s=14.

In order to construct functions that belong to the basis function space, we arbitrarily take a set of coefficient vectors c with structural sparse properties. In this experiment, the first 14 items of the selected coefficient vector are finite, and all other items are 0. The coefficient values are shown in the following table:

TABLE I. THE FIRST 14 VALUES OF THE COEFFICIENT TO BE RECOVERED

c_1	c_2	c_3	c_4	c_5	c_6	c_7
-0.5191	-0.2947	-0.1605	1.0235	0.0159	-0.4827	0.6347
c_8	c_9	c_{10}	c_{11}	c_{12}	c_{13}	c_{14}
-1.3644	0.5986	-0.2065	2.1394	-0.6488	0.4236	0.1179

After obtaining the approximate function $\tilde{f}(t)$, select q points arbitrarily on [-1,1], and compare the error between the original function and the approximate function at these q points. For the sake of simplicity, the selected point is [-1:0.01:1] these 201 points for verification. The function recovery situation is shown in the figure below:

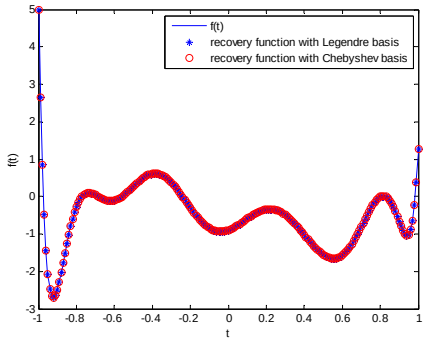


FIGURE I. SCHEMATIC DIAGRAM OF THE LEGENDRE BASIS FUNCTION AND THE CHEBYSHEV BASIS FUNCTION RECOVERY FUNCTION FOR N=512, S=14, M=128

It can be seen from Fig. 1 that when the number of sampling points m is larger than the required number of optimal cases of the algorithm case 50.3897, the function has a very good recovery effect. Since the selection of sampling points $\{t_i\}_{i=1}^{128}$ is random, repeat tests are performed and the experimental results are given in the following two tables.

TABLE II. APPROXIMATE FUNCTION AND ORIGINAL FUNCTION ERROR TABLE AT 201 TEST POINTS

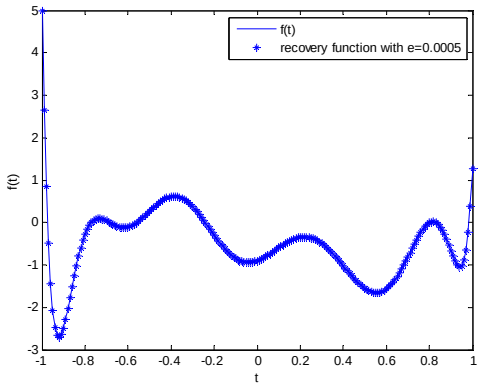
Times	***1***	***2***	***3***	***4***
$L_\infty - error1$	1.3323e-15	1.7764e-15	4.4409e-15	8.8818e-16
$L_2 - error1$	2.9601e-15	3.7105e-15	4.8015e-15	1.8369e-15
Times	***5***	***6***	***7***	***8***
$L_\infty - error1$	1.7764e-15	3.5527e-15	8.8818e-16	2.6645e-14
$L_2 - error1$	4.1711e-15	4.9735e-15	1.9079e-15	2.7184e-14

TABLE III. APPROXIMATION VECTOR AND ORIGINAL VECTOR ERROR TABLE

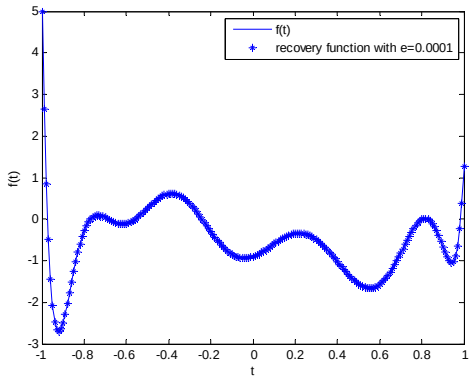
Times	***1***	***2***	***3***	***4***
$L_\infty - error2$	9.9920e-16	8.8818e-16	1.1102e-15	6.6613e-16
$L_2 - error2$	3.6406e-16	4.4715e-16	5.9188e-16	4.1931e-16
Times	***5***	***6***	***7***	***8***
$L_\infty - error2$	8.8818e-16	1.4155e-15	4.4409e-16	3.5527e-15
$L_2 - error2$	5.1658e-16	5.2380e-16	3.1958e-16	2.4327e-15

From Table 2 and Table 3, it is easy to see that both the recovery error of the function and the recovery error of the coefficient are small, and the recovery effect is good.

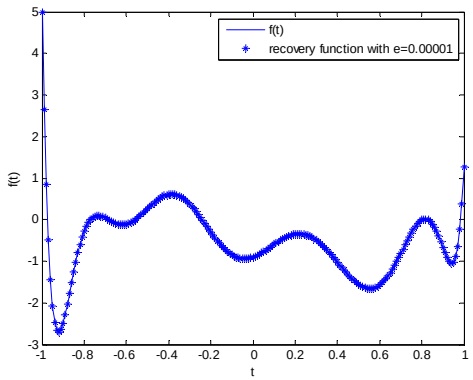
Experiment 2: The sampled value contains noise. In numerical experiments, as in the previous section, the number of basis functions n=512, the number of sampling points m=128, and the sparsity of the coefficient vector s=14. In order to observe the effect of noise on the recovery of the function, the noise values are taken as: $p = e \times rand(m,1)$, where rand(m,1) represents an m-dimensional random vector between 0 and 1, and e represents an arbitrary constant. In this experiment, the selection of coefficient vector c is the same as the selection in experiment 1. Similarly, in order to verify the recovery of the function, select $[-1:0.01:1]$ these 201 points for verification.



(a)



(b)



(c)

FIGURE II. (a)(b)(c) SHOW THE RECOVERY EFFECT OF THE FUNCTION WHEN THE NOISE PARAMETERS E ARE 0.0005, 0.0001, AND 0.00001, RESPECTIVELY.

From Figure 2, it is easy to see that when the noise vector is small, the function recovery effect is very good, because the contrast effect in the image is not very obvious, the following, for the three noise vectors e selected in Figure 2, the function of The recovery error and the recovery error of the coefficient vector are calculated separately, and their corresponding

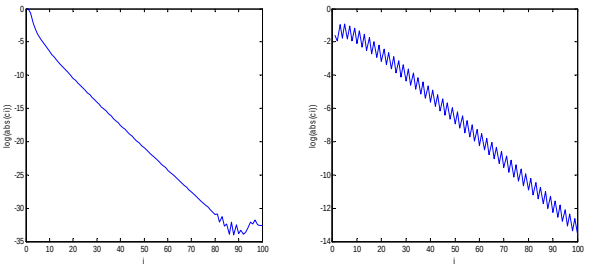
values δ are calculated for the above three different e, and Table 4 is obtained.

TABLE IV. RECOVERY ERROR TABLE WHEN PARAMETER E TAKES DIFFERENT VALUES

error / e	δ	$L_\infty - error2$	$L_2 - error2$	$L_\infty - error1$	$L_2 - error1$
0.0005	0.0032	2.4136e-04	1.0967e-04	7.7612e-04	0.0036
0.0001	6.4894e-04	5.0347e-05	2.1631e-05	1.1069e-04	7.3320e-04
0.00001	6.3849e-05	4.6507e-06	1.8953e-06	7.3323e-06	6.7594e-05

From the above table, it can be seen that when the sparsity of the coefficients is the same, as the parameter δ is gradually reduced, both the recovery error of the coefficients and the recovery error of the functions are reduced.

Second, experiment with the second case. Taking the function $f_1(t) = (1.05+t)^{1/2}$ and $f_2(t) = \dfrac{1+3t}{1+50t^2}$ as examples, if the Legendre polynomial is used as the basis function, the above two functions are first given in the expansion coefficient image of the Legendre basis function.

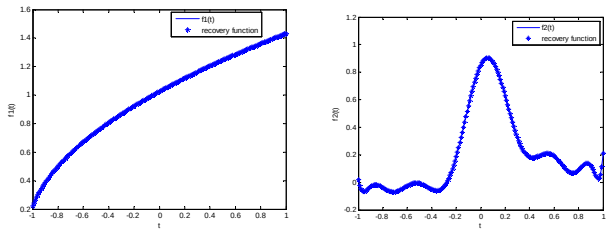


(a) (b)

FIGURE III. (a)(b) SHOW THE FUNCTIONS $f_1(t)$ AND $f_2(t)$ THE FIRST 100 COEFFICIENTS IN THE LEGENDRE BASIS FUNCTION

It can be clearly seen from Figure 3 that when the Legendre polynomial is used as the basis function, the coefficients of the first 100 terms of the expansion of the function $f_1(t)$ and $f_2(t)$ gradually decrease. Therefore, we can think that the expansion coefficients of function $f_1(t)$ and $f_2(t)$ under the Legendre basis function are sparsity. Next, for the case of $f_1(t)$ and $f_2(t)$ in the case of the Legendre base function expansion, numerical experiments were performed on the case where the sampling points contained noise and the sampling points contained no noise.

Experiment 3: When the sampling point is free of noise: In the numerical experiment, the number of basis functions n=1024, the number of sampling points m=128, sparsity s=30. After obtaining the approximate function $\tilde{f}(t)$, select q points arbitrarily on [-1,1], and compare the error between the original function and the approximate function at these q points. For the sake of simplicity, the selected point is [-1:0.01:1]] These 201 points. The experimental results are shown in the figure below:



(a) (b)

FIGURE IV. (a)(b) SHOW THE COMPARISON OF FUNCTION RECOVERY OF $f_1(t)$ AND $f_2(t)$ IN THE LEGENDRE BASIS FUNCTION

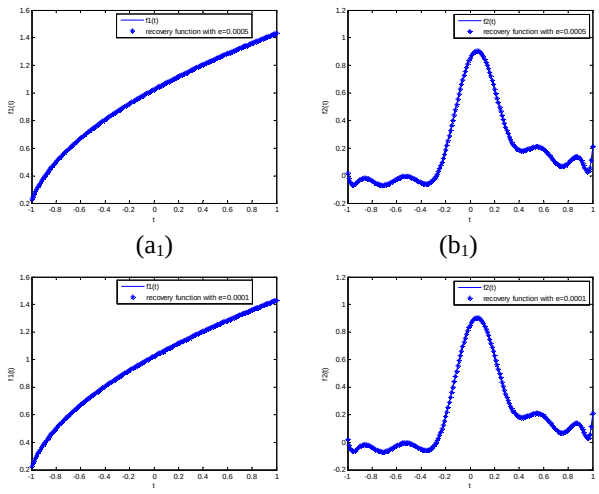
As can be seen from Figure 4, when the first 30 items of the coefficient are selected for function recovery, the function recovery effect is better, and the error is shown in the following table.

TABLE V. FUNCTION RECOVERY ERROR TABLE AT S=30

Error / function	$L_\infty - error2$	$L_2 - error2$	$L_\infty - error1$	$L_2 - error1$
$f_1(t)$	4.4409e-16	7.5624e-16	1.3323e-15	5.6180e-15
$f_2(t)$	2.2204e-16	5.7754e-16	7.2164e-16	2.5715e-15

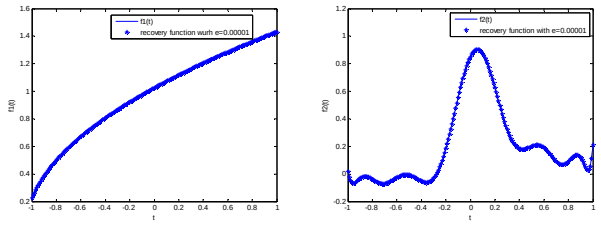
From Table 5, it can be seen that when the sampling point does not contain noise, recovery of a given continuous function is very effective.

Experiment 4: The sampled value contains noise: consistent with the test function in the previous experiment, the function recovery effect under different noises is shown in the figure:



(a_1) (b_1)

(a_2) (b_2)



(a3) (b3)

FIGURE V. FUNCTION RECOVERY DIAGRAM AT DIFFERENT NOISE

Here, (a1)(a2) denotes the recovery of the function when the noise parameter is e=0.0005, and (b1)(b2) represents the recovery of the function when the noise parameter is e=0.0001, respectively; (c1)(c2) respectively represent The recovery of the function when the noise parameter e = 0.00001. (a1)(a2)(a3) and (b1)(b2)(b3) denote functions $f_1(t)$ and $f_2(t)$, respectively.

It is not difficult to see from the above image that when the function is relatively smooth, the function recovered by the l_1 minimization method with inequality constraints is better; when the function changes greatly, the function recovered by the method is in the function The error at the endpoint of the definition domain is larger.

V. CONCLUSION

Through the analysis and experiments of this paper, it can be clearly seen that when the number of sampling points is much less than the number of basis functions in the basis function space, the use of smooth functions in the Fourier series expansion, the coefficient has the characteristics of sparsity, Compressive sensing technology is applied to the problem of function recovery and can better achieve the approximation of the function..

ACKNOWLEDGMENTS

The work is supported by National Natural Science Foundation (No.91630203).

REFERENCES

[1] D. Levin, "Between moving least-squares and moving least-l_1," Bit Numer Math(2015) 55:781-796.

[2] B. Adcock. Infinite-dimensional l_1 minimization and function approximation from pointwise data. arXiv:1503.02352, 2016.

[3] Xu Z Q. "Compressed sensing: a survey(in Chinese)," Sci Sin Math,2012,42(9):865-877.

[4] Yang J, Zhang Y, "Alternating direction algorithm for l_1-problem in compressed sensing," SIAM J Sci Comput, 2011.33:250-278

[5] Cai J, Osher S and Shen Z, "Convergence of the linearized Bregman iteration for l_1-norm minimization," Math Comput, 2009,78: 2127-2136.