Efficient Recovery of Jointly Sparse Vectors

Authored by:

Jieping Ye
Liang Sun
Jun Liu
Jianhui Chen

Abstract

We consider the reconstruction of sparse signals in the multiple measurement vector (MMV) model,in which the signal, represented as a matrix, consists of a set of jointly sparse vectors. MMV is an extension of the single measurement vector (SMV) model employed in standard compressive sensing (CS). Recent theoretical studies focus on the convex relaxation of the MMV problem based on the $(2,1)$-norm minimization, which is an extension of the well-known 1-norm minimization employed in SMV. However, the resulting convex optimization problem in MMV is significantly much more difficult to solve than the one in SMV. Existing algorithms reformulate it as a second-order cone programming (SOCP) or semidefinite programming (SDP), which is computationally expensive to solve for problems of moderate size. In this paper, we propose a new (dual) reformulation of the convex optimization problem in MMV and develop an efficient algorithm based on the prox-method. Interestingly, our theoretical analysis reveals the close connection between the proposed reformulation and multiple kernel learning. Our simulation studies demonstrate the scalability of the proposed algorithm.

1 Paper Body

Compressive sensing (CS), also known as compressive sampling, has recently received increasing attention in many areas of science and engineering [3]. In CS, an unknown sparse signal is reconstructed from a single measurement vector. Recent theoretical studies show that one can recover certain sparse signals from far fewer samples or measurements than traditional methods [4, 8]. In this paper, we consider the problem of reconstructing sparse signals in the multiple measurement vector (MMV) model, in which the signal, represented as a matrix, consists of a set of jointly sparse vectors. MMV is an extension of the single measurement vector (SMV) model employed in standard compressive sensing. The MMV model was motivated by the need to solve the neuromagnetic inverse

problem that arises in Magnetoencephalography (MEG), which is a modality for imaging the brain [7]. It arises from a variety of applications, such as DNA microarrays [11], equalization of sparse communication channels [6], echo cancellation [9], magenetoencephalography [12], computing sparse solutions to linear inverse problems [7], and source localization in sensor networks [17]. Unlike SMV, the signal in the MMV model is represented as a set of jointly sparse vectors sharing their common nonzeros occurring in a set of locations [5, 7]. It has been shown that the additional block-sparse structure can lead to improved performance in signal recovery [5, 10, 16, 21]. Several recovery algorithms have been proposed for the MMV model in the past [5, 7, 18, 24, 25]. Since the sparse representation problem is a combinatorial optimization problem and is in general NP-hard [5], the algorithms in [18, 25] employ the greedy strategy to recover the signal using an iterative scheme. One alternative is to relax it into a convex optimization problem, from which the 1

global optimal solution can be obtained. The most widely studied approach is the one based on the (2, 1)-norm minimization [5, 7, 10]. A similar relaxation technique (via the 1-norm minimization) is employed in the SMV model. Recent studies have shown that most of theoretical results on the convex relaxation of the SMV model can be extended to the MMV model [5], although further theoretical investigation is needed [26]. Unlike the SMV model where the 1-norm minimization can be solved efficiently, the resulting convex optimization problem in MMV is much more difficult to solve. Existing algorithms formulate it as a second-order cone programming (SOCP) or semdefinite programming (SDP) [16] problem, which can be solved using standard software packages such as SeDuMi [23]. However, for problems of moderate size, solving either SOCP or SDP is computationally expensive, which limits their use in practice. In this paper, we derive a dual reformulation of the (2, 1)-norm minimization problem in MMV. More especially, we show that the (2, 1)-norm minimization problem can be reformulated as a min-max problem, which can be solved efficiently via the prox-method with a nearly dimensionindependent convergence rate [19]. Compared with existing algorithms, our algorithm can scale to larger problems while achieving high accuracy. Interestingly, our theoretical analysis reveals the close relationship between the resulting min-max problem and multiple kernel learning [14]. We have performed simulation studies and our results demonstrate the scalability of the proposed algorithm in comparison with existing algorithms. Notations: All matrices are boldface uppercase. Vectors are boldface lowercase. Sets and spaces are denoted with calligraphic letters. The p-norm of the vector v = (v1 , ? ? ? , vd)T ? IRd is defined ?P ?1 d p p as kvkp := . The inner product on IRm?d is defined as hX, Yi = tr(XT Y). For —v — i i=1 matrix A ? IRm?d , we denote by ai and ai the ith row and the ith column of A, respectively. The (r, p)-norm of A is defined as: ?m ! p1 X kAkr,p := kai kpr . (1) i=1

2

The Multiple Measurement Vector Model

In the SMV model, one aims to recover the sparse signal w from a measurement vector b = Aw for a given matrix A [3]. The SMV model can be

extended to the multiple measurement vector (MMV) model, in which the signal is represented as a set of jointly sparse vectors sharing a common set of nonzeros [5, 7]. The MMV model aims to recover the sparse representations for SMVs simultaneously. It has been shown that the MMV model provably improves the standard CS recovery by exploiting the block-sparse structure [10, 21]. Specifically, in the MMV model we consider the reconstruction of the signal represented by a matrix W ? IRd?n , which is given by a dictionary (or measurement matrix) A ? IRm?d and multiple measurement vector B ? IRm?n such that B = AW.

(2)

Each column of A is associated with an atom, and a set of atom is called a dictionary. A sparse representation means that the matrix W has a small number of rows containing nonzero entries. Usually, we have m ? d and d ¿ n. Similar to SMV, we can use kWkp,0 to measure the number of rows in W that contain nonzero entries. Thus, the problem of finding the sparsest representation of the signal W in MMV is equivalent to solving the following problem, a.k.a. the sparse representation problem: (P0) :

min kWkp,0 , W

s.t.

AW = B.

(3)

Some typical choices of p include p = ? and p = 2 [25]. However, solving (P0) requires enumerating all subsets of the set {1, 2, ? ? ? , d}, which is essentially a combinatorial optimization problem and is in general NP-hard [5]. Similar to the use of the 1-norm minimization in the SMV model, one natural alternative is to use kWkp,1 instead of kWkp,0 , resulting in the following convex optimization problem (P1): (P1) : min kWkp,1 , s.t. AW = B. (4) W

2

The relationship between (P0) and (P1) for the MMV model has been studied in [5]. For p = 2, the optimal W is given by solving the following convex optimization problem: 1 min kWk22,1 W 2

s.t. AW = B.

(5)

Existing algorithms formulate Eq. (5) as a second-order cone programming (SOCP) problem or a semidefinite programming (SDP) problem [16]. Recall that the optimizaiton problem in Eq. (5) is equivalent to the following problem by removing the square in the objective: 1 min kWk2,1 W 2

s.t. AW = B.

By introducing auxiliary variable ti (i = 1, ? ? ? , d), this problem can be reformulated in the standard second-order cone programming (SOCP) formulation: d

min

1X ti 2 i=1

s.t.

kWi k2 ? ti , ti ? 0, i = 1, ? ? ? , d,

W,t1 ,??? ,td

(6) AW = B.

Based on this SOCP formulation, it can also be transformed into the standard semidefinite programming (SDP) formulation: d

min

W,t1 ,??? ,td

s.t.

1X ti 2 i=1 ? ? T ti I Wi ? 0, t ? 0, i = 1, ? ? ? , d, i Wi ti

(7) AW = B.

The interior point method [20] and the bundle method [13] can be applied to solve SOCP and SDP. However, they do not scale to problems of moderate size, which limits their use in practice.

3

The Proposed Dual Formulation

In this section we present a dual reformulation of the optimization problem in Eq. (5). First, some preliminary results are summarized in Lemmas 1 and 2: Lemma 1. Let A and X be m-by-d matrices. Then the following holds: ? 1? hA, Xi ? kXk22,1 + kAk22,? . (8) 2 When the equality holds, we have kXk2,1 = kAk2,? . Pm i Proof. It follows from the definition of the (r, p)-norm in Eq. (1) that kXk2,1 = i=1 kx k2 , i k and kAk2,? = max1?i?m ka k2 . Without loss of generality, we assume that ka k2 = max1?i?m kai k2 for 1 ? k ? m . Thus, kAk2,? = kak k2 , and we have hA, Xi =

m X

T

ai xi ?

i=1

?

?

1? k 2 ka k2 + 2

m X

kai k2 kxi k2 ?

i=1

m X

kak k2 kxi k2 = kak k2

i=1

!2 ? ? 1? kAk22,? + kXk22,1 . kxi k2 ? = 2 i=1

?m X

Clearly, the last inequality becomes equality when kXk2,1 = kAk2,? . Lemma 2. Let A and X be defined as in Lemma 1. Then the following holds: ? ? 1 1 2 max hA, Xi ? kXk2,1 = kAk22,? . X 2 2 3

m X i=1

kxi k2

k Proof. Denote the set Q = {k : 1 ? k ? m, ka k2 = max1?i?m kai k2 }. Let {?k }m k=1 be such Pm / Q, ?k ? 0 for k ? Q, and k=1 ?k = 1. Clearly, all inequalities in the proof that ?k = 0 for k ? of Lemma 1 become equalities if and only if we construct the matrix X as follows: ? ?k ak , if k ? Q xk = (9) 0, otherwise.

Thus, the maximum of hA, Xi ? 12 kXk22,1 is 21 kAk22,? , which is achieved when X is constructed as in Eq. (9). Based on the results established in Lemmas 1 and 2, we can derive the dual formulation of the optimization problem in Eq. (5) as follows. First we construct the Lagrangian L: 1 1 kWk22,1 ? hU, AW ? Bi = kWk22,1 ? hU, AWi + hU, Bi. 2 2 The dual problem can be formulated as follows: L(W, U) =

1 max min kWk22,1 ? hU, AWi + hU, Bi. U W 2

(10)

It follows from Lemma 2 that ? ? ? ? 1 1 1 2 2 T min kWk2,1 ? hU, AWi = min kWk2,1 ? hA U, Wi = ? kAT Uk22,? . W W 2 2 2 Note that from Lemma 2, the equality holds if and only if the optimal W? can be represented as W? = diag(?)AT U,

(11)

where ? = [?1 , ? ? ? , ?d]T ? IRd , ?i ? 0 if k(AT U)i k2 = kAT Uk2,? , ?i = 0 if k(AT U)i k2 ¡ Pd kAT Uk2,? , and i=1 ?i = 1. Thus, the dual problem can be simplified into the following form: 1 max ? kAT Uk22,? + hU, Bi. U 2

(12)

Following the definition of the (2, ?)-norm, we can reformulate the dual problem in Eq. (12) as a min-max problem, as summarized in the following theorem: Theorem 1. The optimization problem in Eq. (5) can be formulated equivalently as: ( ) n d X 1X T T max uj bj ? ?i uj Gi uj , (13) Pd min u1 ,??? ,un 2 i=1 i=1 ?i =1,?i ?0 j=1 where the matrix Gi is defined as Gi = ai aTi (1 ? i ? d), and ai is the ith column of A. Proof. Note that kAT Uk22,? can be reformulated as follows: ? ? kAT Uk22,? = max kaTi Uk22 = max {tr(UT ai aTi U)} = max {tr(UT Gi U)} 1?i?d

=

?i ?0,

1?i?d

max P d

i=1

d X ?i =1

1?i?d

?i tr(UT Gi U).

(14)

i=1

Substituting Eq. (14) into Eq. (12), we obtain the following problem: d

1 1X max ? kAT Uk22,? + hU, Bi ? max P min ?i tr(UT Gi U). hU, Bi ? d U U 2 2 i=1 i=1 ?i =1,?i ?0

(15)

Since the Slater?s condition [2] is satisfied, the minimization and maximization in Eq. (15) can be exchanged, resulting in the min-max problem in Eq. (13). Corollary 1. Let (? ? , U?) be the optimal solution to Eq. (13) where ? ? = (?1? , ? ? ? , ?d?)T . If ?i? ¿ 0, then k(AT U?)i k2 = kAT U? k2,? . 4

Based on the solution to the dual problem in Eq. (13), we can construct the optimal solution to the primal problem in Eq. (5) as follows. Let W? be the optimal solution of Eq. (5). It follows from Lemma 2 that we can construct

W? based on AT U? as in Eq. (11). Recall that W? must satisfy the equality constraint AW? = B. The main result is summarized in the following theorem: Theorem 2. Given W? = diag(?)AT U? , where ? = [?1 , ? ? ? , ?d] ? IRd , ?i ? 0, ?i ¿ 0 only if ? ?i Pd k AT U? k2 = kAT U? k2,? , and i=1 ?i = 1. Then, AW? = B if and only if (?, U?) is the optimal solution to the problem in Eq. (13). Proof. First we assume that (?, U?) is the optimal solution to the problem in Eq. (13). It follows that the partial derivative of the objective function with respect to U? in Eq. (13) is 0, that is, B ? Adiag(?)AT U? = 0 ? AW? = B. Next we prove the reverse direction by assuming AW? = B. Since W? = diag(?)AT U? , we have 0 = B ? AW? = B ? Adiag(?1 , ? ? ? , ?d)AT U? . (16) Define the function ?(?1 , ? ? ? , ?d , U) as ( ) d n d X 1X 1X T T T ?(?1 , ? ? ? , ?d , U) = hU, Bi ? ?i tr(U Gi U) = uj bj ? ?i uj Gi uj . 2 i=1 2 i=1 j=1 We consider the function ?(?1 , ? ? ? , ?d , U) with fixed ?i = ?i (1 ? i ? d). Note that this function is concave with respect to U, thus its maximum is achieved when its partial derivative with respect ?? to U is zero. It follows from Eq. (16) that ?U is zero when U = U? . Thus, we have ?U, ?(?1 , ? ? ? , ?d , U) ? ?(?1 , ? ? ? , ?d , U?). ? With a fixed U = U , ?(?1 , ? ? ? , ?d , U?) is a linear combination of ?i (1 ? i ? d) as: d 1X ?(?1 , ? ? ? , ?d , U?) = hU? , Bi ? ?i k(AT U?)i k22 . 2 i=1 By the assumption, we have k(AT U?)i k = kAT U? k2,? , if ?i ¿ 0. Thus, we have d X ?i = 1, ?i ? 0. ?(?1 , ? ? ? , ?d , U?) ? ?(?1 , ? ? ? , ?d , U?), ??1 , ? ? ? , ?d satisfying i=1

Pd

Therefore, for any U, ?1 , ? ? ? , ?d such that i=1 ?i = 1, ?i ? 0, we have ?(?1 , ? ? ? , ?d , U) ? ?(?1 , ? ? ? , ?d , U?) ? ?(?1 , ? ? ? , ?d , U?), (17) which implies that (?1 , ? ? ? , ?d , U?) is a saddle point of the min-max problem in Eq. (13). Thus, (?, U?) is the optimal solution to the problem in Eq. (13). Theorem 2 shows that we can reconstruct the solution to the primal problem based on the solution to the dual problem in Eq. (13). It paves the way for the efficient implementation based on the min-max formulation in Eq.(13). In this paper, the prox-method [19], which is discussed in detail in the next section, is employed to solve the dual problem in Eq. (13). An interesting observation is that the resulting min-max problem in Eq. (13) is closely related to the optimization problem in multiple kernel learning (MKL) [14]. The min-max problem in Eq. (13) can be reformulated as ? n ? X 1 T T max uj bj ? uj Guj , (18) Pd min u1 ,??? ,un 2 i=1 ?i =1,?i ?0 j=1 where the positive semidefinite (kernel) matrix G is constrained as a linear combination of a set of o n Pd T d as G = i=1 ?i Gi . base kernels Gi = ai ai i=1

The formulation in Eq. (18) connects the MMV problem to MKL. Many efficient algorithms [14, 22, 27] have been developed in the past for MKL, which can be applied to solve (13). In [27], an extended level set method was proposed to solve MKL, which was shown to outperform the one based on the semi-infinite linear programming formulation [22]. However, the extended level set method involves a linear programming in each iteration and its theoretical convergence rate of ? O(1/ N) (N denotes the number of iterations) is slower than the proposed algorithm presented in the next section. 5

4

The Main Algorithm

We propose to employ the prox-method [19] to solve the min-max formulation in Eq. (13), which has a differentiable and convex-concave objective function. The algorithm is called ?MMVprox ?. The prox-method is a first-order method [1, 19] which is specialized for solving the saddle point problem and has a nearly dimension-independent convergence rate of O(1/N) (N denotes the number of iterations). We show that each iteration of MMVprox has a low computational cost, thus it scales to large-size problems. The key idea is to convert the min-max problem to the associated variational inequality (v.i.) problem, which is then iteratively solved by a series of v.i. problems. Let z = (?, U). The problem in Eq. (13) is equivalent to the following associated v.i. problem [19]: Find z? = (? ? , U?) ? S : hF (z?), z ? z? i ? 0, ?z ? S, S = X ? Y, where

(19)

? ? ? ?(?, U), ? ?(?, U) (20) ?? ?U is an operator constituted by the gradient of ?(?, ?), X = {? ? IRd : k?k1 = 1, ?i ? 0}, and Y = IRm?n . ?

F (z) =

In solving the v.i. problem in Eq. (19), one key building block is the following projection problem: ? ? 1 ? ? zi , zk22 + h? Pz (? z) = arg min k? z, z (21) ??S 2 z ? U) ? U). ? and z ? Denote (?? , U?) = Pz (? ? = (?, ? = (?, where z z). It is easy to verify that 1 ? ? 2, ?? = arg min k? ? (? ? ?)k 2 ? 2 ??X and

? U? = U ? U.

(22) (23)

Following [19], we present the pseudocode of the proposed MMVprox algorithm in Algorithm 1. In each iteration, we compute the projection (21) so that wt,s is sufficiently close to wt,s?1 (controlled by the parameter ?). It has been shown in [19] that, when ? ? ?12L [L denotes the Lipschitz continuous constant of the operator F (?)], the inner iteration converges within two iterations, i.e., wt,2 = wt,1 always holds. Moreover, Algorithm 1 has a global dimension-independent convergence rate of O(1/N). Algorithm 1 The MMVprox Algorithm Input: A, B, ?, z0 = (?0 , U0), and ? Output: ?, U and W. Step t (t ? 1): Set wt,0 = zt?1 and find the smallest s = 1, 2, . . . such that wt,s = Pzt?1 (?F (wt,s?1)), kwt,s ? wt,s?1 k2 ? ?. Set zt = wt,s Final Step: Set ? =

Pt i=1

t

?i

,U=

Pt i=1

t

Ui

, W = diag(?)AT U.

Time Complexity It costs O(dmn) to evaluate the operator F (?) at a given point. ?? in Eq. (22) involves the Euclidean projection onto the simplex [1], which can be solved in linear time, i.e., in O(d); and U? in Eq. (23) can be analytically computed in O(mn) time. Recall that at each iteration t, the inner

iteration is at most 2. Thus, the time complexity for any given outer iteration is O(dmn). Our analysis shows that MMVprox scales to large-size problems. In comparison, the second-order methods such as SOCP have a much higher complexity per iteration. According to [15], the SOCP in Eq. (6) costs O(d3 (n + 1)3) per iteration. In MMV, d is typically larger than m. In this case, the proposed MMVprox algorithm has a much smaller cost per iteration than SOCP. This explains why MMVprox scales better than SOCP, as shown in our experiments in the next section. 6

Table 1: The averaged recovery results over 10 experiments (d = 100, m = 50, and n = 80). Data set 1 2 3 4 5 6 7 8 9 10 Mean Std

5

p kW ? Wp k2F /(dn) 3.2723e-6 3.4576e-6 2.6971e-6 2.4099e-6 2.9611e-6 2.5701e-6 2.0884e-6 2.3454e-6 2.6807e-6 2.7172e-6 2.7200e-6 4.1728e-7

p kAWp ? Bk2F /(mn) 1.4467e-5 1.8234e-5 1.4464e-5 1.4460e-5 1.4463e-5 1.4459e-5 1.4469e-5 1.4475e-5 1.4461e-5 1.4481e-5 1.4843e-5 1.1914e-6

Experiments

In this section, we conduct simulations to evaluate the proposed MMVprox algorithm in terms of the recovery quality and scalability. Experiment Setup We generated a set of synthetic data sets (by varying the values of m, n, and d) for our experiments: the entries in A ? IRm?d were independently generated from the standard normal distribution N (0, 1); W ? IRd?n (the ground truth of the recovery problems) was generated in two steps: (1) randomly select k rows with nonzero entries; (2) randomly generate the entries of those k rows from N (0, 1). We denote by Wp the solution obtained from the proposed MMVprox algorithm. Ideally, Wp should be close to W. Our experiments were performed on a PC with Intel Core 2 Duo T9500 2.6G CPU and 4G RAM. We employed the optimization package SeDuMi [23] for solving the SOCP formulation. All codes were implemented in Matlab. In all experiments, we terminate MMVprox when the change of the consecutive approximate solutions is less than 1e-6. Recovery Quality In this experiment, we evaluate the recovery quality of the proposed MMVprox algorithm. We applied MMVprox on the data sets of size d = 100, m = 50, n = 80, and reported the averaged experimental resultspover 10 random repetitions. We measured p the recovery quality in terms of the mean squared error: kW ? Wp k2F /(dn). We also reported kAWp ? Bk2F /(mn), which measures the violation of the constraint in Eq. (5). The experimental results are presented in Table 1. We can observe from the table that MMVprox recovers the sparse signal successfully in all cases. Next, we study how the recovery error changes as the sparsity of W varies. Specifically, we applied MMVprox on the data sets of size d = 100, m = 400, and n = 10 with k (the number of nonzero p rows of W) varying from 0.05d to 0.7d, and used kW ? Wp k2F /(dn) as the recovery quality measure. The averaged experimental results over 20 random repetitions are presented in Figure 1. We can observe from the figure that MMVprox works well in all cases, and a larger k (less sparse W) tends to result in a larger recovery error. ?6

p

kW ? Wp k2F /(dn)

2
x 10
1.5 1 0.5 0 0.05
0.2
0.35 k/d
0.5
0.7
Figure 1: The increase of the recovery error as the sparsity level decreases
7

Scalability In this experiment, we study the scalability of the proposed MMVprox algorithm. We generated a collection of data sets by varying m from 10 to 200 with a step size of 10, and setting n = 2m and d = 4m accordingly. We applied SOCP and MMVprox on the data sets and recorded their computation time. The experimental results are presented in Figure 2 (a), where the x-axis corresponds to the value of m, and the y-axis corresponds to log(t), where t denotes the computation time (in seconds). We can observe from the figure that the computation time of both algorithms increases as m increases and SOCP is faster than MMVprox on small problems (m ? 40); when m ¿ 40, MMVprox outperforms SOCP; when the value of m is large (m ¿ 80), the SOCP formulation cannot be solved by SeDuMi, while MMVprox can still be applied. This experimental result demonstrates the good scalability of the proposed MMVprox algorithm in comparison with the SOCP formulation. 10 8
4 SOCP MMVprox
0
log(t)
6
log(t)
SOCP MMVprox
2
4 2
?2 ?4 ?6
0
?8
?2
?10
50
100 m
150
200
(a)
50
100 m
150
200
(b)

Figure 2: Scalability comparison of MMVprox and SOCP: (a) the computation time for both algorithms as the problem size varies; and (b) the average computation time of each iteration for both algorithms as the problem size varies. The x-axis denotes the value of m, and the y-axis denotes the computation time in seconds (in log scale).

To further examine the scalability of both algorithms, we compare the execution time of each iteration for both SOCP and the proposed algorithm. We use the same setting as in the last experiment, i.e., n = 2m, d = 4m, and m ranges from 10 to 200 with a step size of 10. The time comparison of SOCP and MMVprox is presented in Figure 2 (b). We observe that MMVprox has a significantly lower cost than SOCP in each iteration (note that SOCP is not applicable for m ¿ 80). This is consistent with our complexity analysis in Section 4. We can observe from Figure 2 that when m is small, the computation time of SOCP and MMVprox is comparable, although MMVprox is much faster in each iteration. This is because MMVprox is a first-order method, which has a slower convergence rate than the second-order method SOCP. Thus, there is a tradeoff between scalability and convergence rate. Our experiments show the advantage of MMVprox for large-size problems.

6 Conclusions In this paper, we consider the (2, 1)-norm minimization for the reconstruction of sparse signals in the multiple measurement vector (MMV) model, in which the signal consists of a set of jointly sparse vectors. Existing algorithms formulate it as second-order cone programming or semdefinite programming, which is computationally expensive to solve for problems of moderate size. In this paper, we propose an equivalent dual formulation for the (2, 1)-norm minimization in the MMV model, and develop the MMVprox algorithm for solving the dual formulation based on the proxmethod. In addition, our theoretical analysis reveals the close connection between the proposed dual formulation and multiple kernel learning. Our simulation studies demonstrate the effectiveness of the proposed algorithm in terms of recovery quality and scalability. In the future, we plan to compare existing solvers for multiple kernel learning [14, 22, 27] with the proposed MMVprox algorithm. In addition, we plan to examine the efficiency of the prox-method for solving various MKL formulations.

Acknowledgements This work was supported by NSF IIS-0612069, IIS-0812551, CCF-0811790, NIH R01-HG002516, NGA HM1582-08-1-0016, and NSFC 60905035.
8

2 References

[1] A. Ben-Tal and A. Nemirovski. Non-Euclidean restricted memory level method for large-scale convex optimization. Mathematical Programming, 102(3):407?56, 2005. [2] S. Boyd and L. Vandenberghe. Convex Optimization. Cambridge University Press, Cambridge, UK, 2004. [3] E. Cand‘es. Compressive sampling. In International Congress of Mathematics, number 3, pages 1433? 1452, Madrid, Spain, 2006. [4] E. Cand‘es, J. Romberg, and T. Tao. Robust un-

certainty principles: exact signal reconstruction from highly incomplete frequency information. IEEE Transactions on Information Theory, 52(2):489?509, 2006. [5] J. Chen and X. Huo. Theoretical results on sparse representations of multiple-measurement vectors. IEEE Transactions on Signal Processing, 54(12):4634?4643, 2006. [6] S.F. Cotter and B.D. Rao. Sparse channel estimation via matching pursuit with application to equalization. IEEE Transactions on Communications, 50(3):374?377, 2002. [7] S.F. Cotter, B.D. Rao, Kjersti Engan, and K. Kreutz-Delgado. Sparse solutions to linear inverse problems with multiple measurement vectors. IEEE Transactions on Signal Processing, 53(7):2477?2488, 2005. [8] D.L. Donoho. Compressed sensing. IEEE Transactions on Information Theory, 52(4):1289?1306, 2006. [9] D.L. Duttweiler. Proportionate normalized least-mean-squares adaptation in echo cancelers. IEEE Transactions on Speech and Audio Processing, 8(5):508?518, 2000. [10] Y.C. Eldar and M. Mishali. Robust recovery of signals from a structured union of subspaces. To Appear in IEEE Transactions on Information Theory, 2009. [11] S. Erickson and C. Sabatti. Empirical bayes estimation of a sparse vector of gene expression changes. Statistical Applications in Genetics and Molecular Biology, 4(1):22, 2008. [12] I.F. Gorodnitsky, J.S. George, and B.D. Rao. Neuromagnetic source imaging with focuss: a recursive weighted minimum norm algorithm. Electroencephalography and Clinical Neurophysiology, 95(4):231?251, 1995. [13] H. Jean-Baptiste and C. Lemarechal. Convex Analysis and Minimization Algorithms I: Fundamentals (Grundlehren Der Mathematischen Wissenschaften). Springer, Berlin, 1993. [14] G.R.G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M.I. Jordan. Learning the kernel matrix with semidefinite programming. Jouranl of Machine Learning Research, 5:27?72, 2004. [15] M. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. Linear Algebra and its Applications, 284(1-3):193?228, 1998. [16] F. Parvaresh M. Stojnic and B. Hassibi. On the reconstruction of block-sparse signals with an optimal number of measurements. CoRR, 2008. [17] D. Malioutov, M. Cetin, and A. Willsky. Source localization by enforcing sparsity through a laplacian. In IEEE Workshop on Statistical Signal Processing, pages 553?556, 2003. [18] M. Mishali and Y.C. Eldar. Reduce and boost: Recovering arbitrary sets of jointly sparse vectors. IEEE Transactions on Signal Processing, 56(10):4692?4702, 2008. [19] A. Nemirovski. Prox-method with rate of convergence o(1/t) for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. SIAM Journal on Optimization, 15(1):229?251, 2005. [20] Y.E. Nesterov and A.S. Nemirovskii. Interior-point Polynomial Algorithms in Convex Programming. SIAM Publications, Philadelphia, PA, 1994. [21] M. Duarte R.G. Baraniuk, V. Cevher and C. Hegde. Model-based compressive sensing. Submitted to IEEE Transactions on Information Theory, 2008. [22] S. Sonnenburg, G. R?atsch, C. Sch?afer, and B. Sch?olkopf. Large scale multiple kernel learning. Journal of Machine Learning Research, 7:1531?1565, 2006. [23] J.F. Sturm. Using sedumi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optimization Methods and Software, 11(12):625?653, 1999. [24] J.A. Tropp. Algorithms for simultaneous sparse approximation. Part II:

Convex relaxation. Signal Processing, 86(3):589?602, 2006. [25] J.A. Tropp, A.C. Gilbert, and M.J. Strauss. Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit. Signal Processing, 86(3):572?588, 2006. [26] E. van den Berg and M. P. Friedlander. Joint-sparse recovery from multiple measurements. Technical Report, Department of Computer Science, University of British Columbia, 2009. [27] Z. Xu, R. Jin, I. King, and M.R. Lyu. An extended level method for efficient multiple kernel learning. In Advances in Neural Information Processing Systems, pages 1825?1832, 2008.

9