Multivariate Regression with Calibration

Authored by:

Han Liu
Tuo Zhao
Lie Wang

Abstract

We propose a new method named calibrated multivariate regression (CMR) for fitting high dimensional multivariate regression models. Compared to existing methods, CMR calibrates the regularization for each regression task with respect to its noise level so that it is simultaneously tuning insensitive and achieves an improved finite-sample performance. Computationally, we develop an efficient smoothed proximal gradient algorithm which has a worst-case iteration complexity $O(1/epsilon)$, where $epsilon$ is a pre-specified numerical accuracy. Theoretically, we prove that CMR achieves the optimal rate of convergence in parameter estimation. We illustrate the usefulness of CMR by thorough numerical simulations and show that CMR consistently outperforms other high dimensional multivariate regression methods. We also apply CMR on a brain activity prediction problem and find that CMR is as competitive as the handcrafted model created by human experts.

1 Paper Body

Given a design matrix X 2 Rn?d and a response matrix Y 2 Rn?m , we consider a multivariate linear model Y = XB0 + Z, where B0 2 Rd?m is an unknown regression coefficient matrix and Z 2 Rn?m is a noise matrix [1]. For a matrix A = [Ajk] 2 Rd?m , we denote Aj? = (Aj1 , ..., Ajm) 2 Rm and A?k = (A1k , ..., Adk)T 2 Rd to be its j th row and k th column respectively. We assume that all Zi? ?s are independently sampled from an m-dimensional Gaussian distribution with mean 0 and covariance matrix ? 2 Rm?m .

We can represent the multivariate linear model as an ensemble of univariate linear regression models: Y?k = XB0?k +Z?k , k = 1, ..., m. Then we get a multi-task learning problem [3, 2, 26]. Multi-task learning exploits shared common structure across tasks to obtain improved estimation performance. In the past decade, significant progress has been made towards designing a variety of modeling assumptions for multivariate regression. A popular assumption is that all the regression tasks share a common sparsity pattern, i.e., many B0j? ?s

are zero vectors. Such a joint sparsity assumption is a natural extension of that for univariate linear regressions. Similar to the L1 -regularization used in Lasso [23], we can adopt group regularization to obtain a good estimator of B0 [25, 24, 19, 13]. Besides the aforementioned approaches, there are other methods that aim to exploit the covariance structure of the noise matrix Z [7, 22]. For ? The authors are listed in alphabetical order. This work is partially supported by the grants NSF IIS1408910, NSF IIS1332109, NSF Grant DMS-1005539, NIH R01MH102339, NIH R01GM083084, and NIH R01HG06841. ? Tuo Zhao is also affiliated with Department of Operations Research and Financial Engineering at Princeton University.

1

instance, [22] assume that all Zi? ?s follow a multivariate Gaussian distribution with a sparse inverse covariance matrix ? = ? 1 . They propose an iterative algorithm to estimate sparse B0 and ? by maximizing the penalized Gaussian log-likelihood. Such an iterative procedure is effective in many applications, but the theoretical analysis is difficult due to its nonconvex formulation. In this paper, we assume an uncorrelated structure for the noise matrix Z, i.e., ? = 2 2 diag(12 , 22 , . . . , m 1 , m). Under this setting, we can efficiently solve the resulting estimation problem with a convex program as follows b = argmin p1 ——Y XB——2 + ——B——1,p , B (1.1) F n B qP 2 where ¿ 0 is a tuning parameter, and ——A——F = j,k Ajk is the Frobenius norm of a maP d qP m 2 trix A. Popular choices of p include p = 2 and p = 1: ——B——1,2 = j=1 k=1 Bjk and Pd ——B——1,1 = j=1 max1?k?m —Bjk —. Computationally, the optimization problem in (1.1) can be efficiently solved by some first order algorithms [11, 12, 4].

The problem with the uncorrelated noise structure is amenable to statistical analysis. Under suitable conditions on the noise and design matrices, let max = maxk k , if we choose = p b in (1.1) achieves the opti2c ? max log d + m1 1/p , for some c ¿ 1, then the estimator B mal rates of convergence1 [13], i.e., there exists some universal constant C such that with high probability, we have ! r r 1 b s log d sm1 2/p 0 p ——B B ——F ? C ? max + , nm n m where s is the number of rows with non-zero entries in B0 . However, the estimator in (1.1) has two drawbacks: (1) All the tasks are regularized by the same tuning parameter , even though different tasks may have different k ?s. Thus more estimation bias is introduced to the tasks with smaller k ?s to compensate the tasks with larger k ?s. In another word, these tasks are not calibrated. (2) The tuning parameter selection involves the unknown parameter max . This requires tuning the regularization parameter over a wide range of potential values to get a good finite-sample performance.

To overcome the above two drawbacks , we formulate a new convex program named calibrated multivariate regression (CMR). The CMR estimator is defined to be the solution of the following convex program: b = argmin ——Y XB——2,1 + ——B——1,p , B (1.2) P qP

B

where ——A——2,1 = k is the nonsmooth L2,1 norm of a matrix A = [Ajk] 2 Rd?m . This is a multivariate extension of the square-root Lasso [5].

Similar to the square-root Lasso, the tuning parameter selection of CMR does not involve max . Moreover, the L2,1 loss function can be viewed as a special example of the weighted least square loss, which calibrates each regression task (See more details in ?2). Thus CMR adapts to different k ?s and achieves better finite-sample performance than the ordinary multivariate regression estimator (OMR) defined in (1.1). 2 j Ajk

Since both the loss and penalty functions in (1.2) are nonsmooth, CMR is computationally more challenging than OMR. To efficiently solve CMR, we propose a smoothed proximal gradient (SPG) algorithm with an iteration complexity O(1/?), where ? is the pre-specified accuracy of the objective value [18, 4]. Theoretically, we provide sufficient conditions under which CMR achieves the optimal rates of convergence in parameter estimation. Numerical experiments on both synthetic and real data show that CMR universally outperforms existing multivariate regression methods. For a brain activity prediction task, prediction based on the features selected by CMR significantly outperforms that based on the features selected by OMR, and is even competitive with that based on the handcrafted features selected by human experts. Notations: Given a vector v = (v1 , . . . , vd)T 2 Rd , for 1 ? p ? 1, we define the Lp -vector ?P ?1/p d p norm of v as ——v——p = if 1 ? p ¡ 1 and ——v——p = max1?j?d —vj — if p = 1. j=1 —vj — 1

The rate of convergence is optimal when p = 2, i.e., the regularization function is ——B——1,p

2

Given two matrices A = [Ajk] and C = [Cjk] 2 Rd?m , we define the inner product of A Pd Pm and C as hA, Ci = j=1 k=1 Ajk Cjk = tr(AT C), where tr(A) is the trace of a matrix A. We use A?k = (A1k , ..., Adk)T and Aj? = (Aj1 , ..., Ajm) to denote the k th column and j th row of A. Let S be some subspace of Rd?m , we use AS to denote the projection of A onto S: 2 AS = argmin p C2S ——C A——F . Moreover, we define the Frobenius and spectral norms of A as ——A——F = hA, Ai and ——A——2 = 1 (A), 1 (A) the largest singular value of A. In addition, Pis m we define the matrix block norms as ——A——2,1 = k=1 ——A?k ——2 , ——A——2,1 = max1?k?m ——A?k ——2 , Pd ——A——1,p = j=1 ——Aj? ——p , and ——A——1,q = max1?j?d ——Aj? ——q , where 1 ? p ? 1 and 1 ? q ? 1. It is easy to verify that ——A——2,1 is the dual norm of ——A——2,1 . Let 1/1 = 0, then if 1/p + 1/q = 1, ——A——1,q and ——A——1,p are also dual norms of each other.

2

Method

We solve the multivariate regression problem by the following convex program, b = argmin ——Y XB——2,1 + ——B——1,p . B

(2.1)

B

The only difference between (2.1) and (1.1) is that we replace the L2 -loss function by the nonsmooth L2,1 -loss function. The L2,1 -loss function can be viewed as a special example of the weighted square loss function. More specifically, we consider the following optimization problem, m X 1 b = argmin

p ——Y?k XB?k ——22 + ——B——1,p , B (2.2) k n B k=1

where k n is a weight assigned to calibrate the k th regression task. Without prior knowledge on k ?s, we use the following replacement of k ?s, 1 ek = p ——Y?k XB?k ——2 , k = 1, ..., m. (2.3) n By plugging (2.3) into the objective function in (2.2), we get (2.1). In another word, CMR calibrates different tasks by solving a penalized weighted least square program with weights defined in (2.3). 1p

The optimization problem in (2.1) can be solved by the alternating direction method of multipliers (ADMM) with a global convergence guarantee [20]. However, ADMM does not take full advantage of the problem structure in (2.1). For example, even though the L2,1 norm is nonsmooth, it is nondifferentiable only when a task achieves exact zero residual, which is unlikely in applications. In this paper, we apply the dual smoothing technique proposed by [18] to obtain a smooth surrogate function so that we can avoid directly evaluating the subgradient of the L2,1 loss function. Thus we gain computational efficiency like other smooth loss functions. We consider the Fenchel?s dual representation of the L2,1 loss: ——Y

XB——2,1 =

max

——U——2,1 ?1

hU, Y

XBi.

(2.4)

Let ? ¿ 0 be a smoothing parameter. The smooth approximation of the L2,1 loss can be obtained by solving the following optimization problem ? ——Y XB——? = max hU, Y XBi ——U——2F , (2.5) 2 ——U——2,1 ?1 where ——U——2F is the proximity function. Due to the fact that ——U——2F ? m——U——22,1 , we obtain the following uniform bound by combing (2.4) and (2.5), m? ——Y XB——2,1 ? ——Y XB——? ? ——Y XB——2,1 . (2.6) 2 From (2.6), we see that the approximation error introduced by the smoothing procedure can be controlled by a suitable ?. Figure 2.1 shows several two-dimensional examples of the L2 norm b B with smoothed by different ??s. The optimization problem in (2.5) has a closed form solution U B b U?k = (Y?k XB?k)/ max {——Y?k XB?k ——2 , ?}. The next lemma shows that ——Y

XB——? is smooth in B with a simple form of gradient. 3

(a) ? = 0

(b) ? = 0.1

(c) ? = 0.25

(d) ? = 0.5

Figure 2.1: The L2 norm (? = 0) and its smooth surrogates with ? = 0.1, 0.25, 0.5. A larger ? makes the approximation more smooth, but introduces a larger approximation error. Lemma 2.1. For any ? ¿ 0, ——Y XB——? is a convex and continuously differentiable function in B. In addition, G? (B)?the gradient of ——Y XB——? w.r.t. B?has the form ? ? b B , Y XBi + ?——U b B ——2 /2 @ hU F b B. G? (B) = = XT U (2.7) @B Moreover, let = ——X——22 , then we have that G? (B) is Lipschitz continuous in B with the

Lipschitz constant /?, i.e., for any B0 , B00 2 Rd?m , b B0 U b B00 i——F ? 1 ——XT X(B0 B00)——F ? ——B0 B00 ——F . ——G? (B0) G? (B00)——F = ——hX, U ? ?

Lemma 2.1 is a direct result of Theorem 1 in [18] and implies that ——Y XB——? has good computational structure. Therefore we apply the smooth proximal gradient algorithm to solve the smoothed version of the optimization problem as follows, e = argmin ——Y XB——? + ——B——1,p . B (2.8) B

We then adopt the fast proximal gradient algorithm to solve (2.8) [4]. To derive the algorithm, we first define three sequences of auxiliary variables {A(t) }, {V(t) }, and {H(t) } with A(0) = H(0) = V(0) = B(0) , a sequence of weights {?t = 2/(t + 1)}, and a nonincreasing sequence of step-sizes {?t ¿ 0}. For simplicity, we can set ?t = ?/ . In practice, we use the backtracking line search to dynamically adjust ?t to boost the performance. At the tth iteration, we first take V(t) = (1 ?t)B(t 1) + ?t A(t 1) . We then consider a quadratic approximation of ——Y XH——? as ? ? 1 Q H, V(t) , ?t = ——Y XV(t) ——? + hG? (V(t)), H V(t) i + ——H V(t) ——2F . 2?t e (t) = V(t) ?t G? (V(t)), we take Consequently, let H ? ? 1 e (t) ——2 + ——H——1,p . (2.9) H(t) = argmin Q H, V(t) , ?t + ——H——1,p = argmin ——H H F 2?t H H n o (t) e j? ? max 1 ?t /——H e j? ——2 , 0 . More When p = 2, (2.9) has a closed form solution Hj? = H details about other choices of p in the L1,p norm can be found in [11] and [12]. To ensure that the objective value is nonincreasing, we choose B(t) =

argmin B2{H(t) , B(t

1) }

At last, we take A(t) = B(t 1) + ?1t (H(t) B(t where ” is the stopping precision.

——Y

1)

XB——? + ——B——1,p .

(2.10)

). The algorithm stops when ——H(t) V(t) ——F ? ”,

The numerical rate of convergence of the proposed algorithm with respect to the original optimization problem (2.1) is presented in the following theorem. p b F /? Theorem 2.2. Given a pre-specified accuracy ? and let ? = ?/m, after t = 2 m ——B(0) B—— (t) (t) b 2,1 + ——B—— b 1,p + ?. 1 = O (1/?) iterations, we have ——Y XB ——2,1 + ——B ——1,p ? ——Y XB——

The proof of Theorem 2.2 is provided in Appendix A.1. This result achieves the minimax optimal rate of convergence over all first order algorithms [18]. 4

3

Statistical Properties

For notational simplicity, we define a re-scaled noise matrix W = [Wik] 2 Rn?m with Wik = Zik / k , where EZ2ik = k2 . Thus W is a random matrix with all entries having mean 0 and variance 1. We define G0 to be the gradient of ——Y XB——2,1 at B = B0 . It is easy to see that

XT Z?k XT W?k k XT W?k = = ——Z?k ——2 ——W?k k ——2 ——W?k ——2 does not depend on the unknown quantities k for all k = 1, ..., m. G0?k

works as an important pivotal in our analysis. Moreover, our analysis exploits the decomposability of the L1,p norm [17]. More specifically, we assume that B0 has s rows with all zero entries and define G0?k =

S = C 2 Rd?m — Cj? = 0 for all j such that B0j? = 0 ,

N = C2R

d?m

— Cj? = 0 for all j such that

B0j?

6= 0 .

(3.1) (3.2)

Note that we have B 2 S and the L1,p norm is decomposable with respect to the pair (S, N), i.e., ——A——1,p = ——AS ——1,p + ——AN ——1,p . The next lemma shows that when is suitably chosen, the solution to the optimization problem in (2.1) lies in a restricted set. b be the optimum to (2.1), and 1/p + 1/q = 1. We denote the Lemma 3.1. Let B0 2 S and B 0 b b estimation error as = B B . If c——G0 ——1,q for some c ¿ 1, we have ? c+1 b 2 Mc := 2 Rd?m — —— N ——1,p ? —— S ——1,p . (3.3) c 1 0

The proof of Lemma 3.1 is provided in Appendix B.1. To prove the main result, we also need to assume that the design matrix X satisfies the following condition. Assumption 3.1. Let B0 2 S, then there exist positive constants ? and c ¿ 1 such that ——X ——F p ?? min . n—— ——F 2Mc {0} Assumption 3.1 is the generalization of the restricted eigenvalue conditions for analyzing univariate sparse linear models [17, 15, 6], Many common examples of random design satisfy this assumption [13, 21]. Note that Lemma 3.1 is a deterministic result of the CMR estimator for a fixed . Since G is essentially a random matrix, we need to show that cR? (G0) holds with high probability to deliver a concrete rate of convergence for the CMR estimator in the next theorem. p Theorem 3.2. We assume that each column of X is normalized as m1/2 1/p kX?j k2 = n for all j = 1, ..., d. Then for some universal constant c0 and large enough n, taking p 2c(m1 1/p + log d) p = , (3.4) 1 c0 with probability at least 1 2 exp(2 log d) 2 exp nc20 /8 + log m , we have ! r r r 1 b 16c max 1 + c0 sm1 2/p s log d 0 p ——B B ——F ? 2 + . ? (c 1) 1 c0 n nm m

The proof of Theorem 3.2 is provided in Appendix B.2.pNote that when we choose p = 2, the column normalization condition is reduced to kX?j k2 = n. Meanwhile, the corresponding error bound is further reduced to ! r r 1 b s s log d 0 p ——B B ——F = OP + , n nm m

which achieves the minimax optimal rate of convergence presented in [13]. See Theorem 6.1 in [13] for more technical details. From Theorem 3.2, we see that CMR achieves the same rates of convergence as the noncalibrated counterpart, but the tuning parameter in (3.4) does not involve k ?s. Therefore CMR not only calibrates all the regression tasks, but also makes the tuning parameter selection insensitive to max . 5

4

Numerical Simulations

To compare the finite-sample performance between the calibrated multivariate regression (CMR) and ordinary multivariate regression (OMR), we generate

a training dataset of 200 samples. More specifically, we use the following data generation scheme: (1) Generate each row of the design matrix Xi? , i = 1, ..., 200, independently from a 800-dimensional normal distribution N (0, ?) where ?jj = 1 and ?j‘ = 0.5 for all ‘ 6= j.(2) Let k = 1, . . . , 13, set the regression coefficient matrix B0 2 R800?13 as B01k = 3, B02k = 2, B04k = 1.5, and B0jk = 0 for all j 6= 1, 2, 4. (3) Generate the random noise matrix Z = WD, where W 2 R200?13 with all entries of W are independently generated from N (0, 1), and D is either of the following matrices ? ? DI = max ? diag 20/4 , 2 1/4 , ? ? ? , 2 11/4 , 2 12/4 2 R13?13 DH = max ? I 2 R13?13 . We generate a validation set of 200 samples for the regularization parameter selection and a testing set of 10,000 samples to evaluate the prediction accuracy.

In numerical experiments, we set max = 1, 2, and 4 to illustrate the tuning insensitivity of CMR. The regularization parameter of both CMR and p OMR ispchosen over a grid ? = 240/4 0 , 239/4 0 , ? ? ? , 2 17/4 0 , 2 18/4 0 , where 0 = log d + m. The optimal regulare X eB b ——2 , where ization parameter b is determined by the prediction error as b = argmin 2? ——Y F b denotes the obtained estimate using the regularization parameter , and X e and Y e denote the B design and response matrices of the validation set. Since the noise level k ?s are different in regression tasks, we adopt the following three crite1 b F , Adj. Pre. Err. = ria to evaluate the empirical performance: Pre. Err. = 10000 ——Y XB—— 1 1 0 2 1 2 b b B —— , where X and Y denotes the design XB)D ——F , and Est. Err. = m ——B F 10000m ——(Y and response matrices of the testing set.

All simulations are implemented by MATLAB using a PC with Intel Core i5 3.3GHz CPU and 16GB memory. CMR is solved by the proposed smoothing proximal gradient algorithm, where we set the stopping precision ” = 10 4 , the smoothing parameter ? = 10 4 . OMR is solved by the monotone fast proximal gradient algorithm, where we set the stopping precision ” = 10 4 . We set p = 2, but the extension to arbitrary p ¿ 2 is straightforward. We first compare the smoothed proximal gradient (SPG) algorithm with the ADMM algorithm (the detailed derivation of ADMM can be found in Appendix A.2). We adopt the backtracking line search to accelerate both algorithms with a shrinkage parameter ? = 0.8. We set max = 2 for the adopted multivariate linear models. We conduct 200 simulations. The results are presented in Table 4.1. The SPG and ADMM algorithms attain similar objective values, but SPG is up to 4 times faster than ADMM. Both algorithms also achieve similar estimation errors. We then compare the statistical performance between CMR and OMR. Tables 4.2 and 4.3 summarize the results averaged over 200 replicates. In addition, we also present the results of the oracle estimator, which is obtained by solving (2.2), since we know the true values of k ?s. Note that the oracle estimator is only for comparison purpose, and it is not a practical estimator. Since CMR calibrates the regularization for each task with respect to k , CMR universally outperforms OMR, and achieves almost the same performance as the oracle estimator when we adopt the scale matrix DI to generate the random noise. Meanwhile, when we adopt the scale matrix DH , where all k ?s are the same, CMR and OMR achieve similar performance. This further implies that CMR

can be a safe replacement of OMR for multivariate regressions. In addition, we also examine the optimal regularization parameters for CMR and OMR over all replicates. We visualize the distribution of all 200 selected b?s using the kernel density estimator. In particular, we adopt the Gaussian kernel, and the kernel bandwidth is selected based on the 10fold cross validation. Figure 4.1 illustrates the estimated density functions. The horizontal axis ? ? b corresponds to the rescaled regularization parameter as log plog d+pm . We see that the optimal regularization parameters of OMR significantly vary with different max . In contrast, the optimal regularization parameters of CMR are more concentrated. This is inconsistent with our claimed tuning insensitivity. 6

Table 4.1: Quantitive comparison of the computational performance between SPG and ADMM with the noise matrices generated using DI . The results are averaged over 200 replicates with standard errors in parentheses. SPG and ADMM attain similar objective values, but SPG is up to about 4 times faster than ADMM.

2
0
0
0.5
0
Algorithm
Timing (second)
Obj. Val.
Num. Ite.
Est. Err.
SPG ADMM
2.8789(0.3141) 8.4731(0.8387)
508.21(3.8498) 508.22(3.7059)
493.26(52.268) 437.7(37.4532)
0.1213(0.0286) 0.1215(0.0291)
SPG ADMM
3.2633(0.3200) 11.976(1.460)
370.53(3.6144) 370.53(3.4231)
565.80(54.919) 600.94(74.629)
0.0819(0.0205) 0.0822(0.0233)
SPG ADMM
3.7868(0.4551) 18.360(1.9678)
297.24(3.6125) 297.25(3.3863)
652.53(78.140) 1134.0(136.08)
0.1399(0.0284) 0.1409(0.0317)

Table 4.2: Quantitive comparison of the statistical performance between CMR and OMR with the noise matrices generated using DI . The results are averaged over 200 simulations with the standard errors in parentheses. CMR universally outperforms OMR, and achieves almost the same performance as the oracle estimator. Method

Pre. Err.

Adj. Pre.Err
Est. Err.
1
Oracle CMR OMR
5.8759(0.0834) 5.8761(0.0673) 5.9012(0.0701)
1.0454(0.0149) 1.0459(0.0123) 1.0581(0.0162)
0.0245(0.0086) 0.0249(0.0071) 0.0290(0.0091)
2
Oracle CMR OMR
23.464(0.3237) 23.465(0.2598) 23.580(0.2832)
1.0441(0.0148) 1.0446(0.0121) 1.0573(0.0170)
0.0926(0.0342) 0.0928(0.0279) 0.1115(0.0365)
4
Oracle CMR OMR
93.532(0.8843) 93.542(0.9794) 94.094(1.0978)
1.0418(0.0962) 1.0421(0.0118) 1.0550(0.0166)
0.3342(0.1255) 0.3346(0.1063) 0.4125(0.1417)
max

Table 4.3: Quantitive comparison of the statistical performance between CMR and OMR with the noise matrices generated using DH . The results are averaged over 200 simulations with the standard errors in parentheses. CMR and OMR achieve similar performance. Method

Pre. Err.
Adj. Pre.Err
Est. Err.
1
CMR OMR
13.565(0.1408) 13.697(0.1554)
1.0435(0.0108) 1.0486(0.0142)
0.0599(0.0164) 0.0607(0.0128)
2
CMR OMR
54.171(0.5771) 54.221(0.6173)
1.0418(0.0110) 1.0427(0.0118)
0.2252(0.0649) 0.2359(0.0821)
4
CMR OMR
215.98(2.104) 216.19(2.391)
1.0384(0.0101) 1.0394(0.0114)
0.80821(0.25078) 0.81957(0.31806)
max
1.4
Oracle(1) Oracle(2) Oracle(4) CMR(1) CMR(2) CMR(4) OMR(1) OMR(2) OMR(4)
1.2
1

0.8
0.6
1.4
1
0.8
0.6
0.4
0.4
0.2
0.2
0 ?2
?1.5
?1
?0.5
0
0.5
1
1.5
2
CMR(1) CMR(2) CMR(4) OMR(1) OMR(2) OMR(4)
1.2
0 ?2
2.5
(a) The noise matrices are generated using DI
?1.5
?1
?0.5
0
0.5
1
1.5
2
2.5
(b) The noise matrices are generated using DH

Figure 4.1: The distributions of the selected regularization parameters using the kernel density estimator. The numbers in the parentheses are max ?s. The optimal regularization parameters of OMR are spreader with different max than those of CMR and the oracle estimator. 7

5

Real Data Experiment

We apply CMR on a brain activity prediction problem which aims to build a parsimonious model to predict a person?s neural activity when seeing a stimulus word. As is illustrated in Figure 5.1, for a given stimulus word, we first encode it into an intermediate semantic feature vector using some corpus statistics. We then model the brain?s neural activity pattern using CMR. Creating such a predictive model not only enables us to explore new analytical tools for the

fMRI data, but also helps us to gain deeper understanding on how human brain represents knowledge [16]. Predict fMRI brain activity patterns in response to text stimulus 89/:4:2%,-.&2

!"#$%)/01'2 !"#$%&'()*'

%0334'&

(Mitchell et al., Science,2008)

!"#$%0334'& stimulus word

!"#$%50//'.&

%50//'.&

predicted activities for "apple"

"apple"

!"#$%6)*7*4'&

%6)*7*4'&

+',%,-.&

?

Model

intermediate semantic features

Standard solution

Our solution

Linear models (More restrictive)

(Less restrictive)

(a) illustration of the data collection procedure Nonlinear models .

mapping learned from fMRI data

(b) model for predicting fMRI brain activity pattern

Figure 5.1: An illustration of the fMRI brain activity prediction problem [16].

(a) To collect the ;5'%'¡3'.)/'+=2%0.'%*-+&:*='&%)+%¿")=*5'44%'=%0?%8*)'+*'%@AB data, a human participant sees a sequence of English words and their images. The corresponding fMRI images are recorded to represent the brain activity patterns; (b) To build a predictive model, each stimulus word is encoded into intermediate semantic features (e.g. the co-occurrence statistics of this stimulus word in a large text corpus). These intermediate features can then be used to predict the brain activity pattern. Our experiments involves 9 participants, and Table 5.1 summarizes the prediction performance of different methods on these participants. We see that the prediction based on the features selected by CMR significantly outperforms that based on the features selected by OMR, and is as competitive as that based on the handcrafted features selected by human experts. But due to the space limit, we present the details of the real data experiment in the technical report version. Table 5.1: Prediction accuracies of different methods (higher is better). CMR outperforms OMR for 8 out of 9 participants, and outperforms the handcrafted basis words for 6 out of 9 participants Method

P. 1

CMR 0.840 OMR 0.803 Handcraft 0.822

6

P. 2

P. 3

P. 4
P. 5
P. 6
P. 7
P. 8
P. 9
0.794 0.789 0.776
0.861 0.801 0.773
0.651 0.602 0.727
0.823 0.766 0.782
0.722 0.623 0.865
0.738 0.726 0.734
0.720 0.749 0.685
0.780 0.765 0.819
Discussions

A related method is the square-root sparse multivariate regression [8]. They solve the convex program with the Frobenius loss function and L1,p regularization function b = argmin ——Y B B

XB——F + ——B——1,p .

(6.1)

The Frobenius loss function in (6.1) makes the regularization parameter selection independent of max , but it does not calibrate different regression tasks. Note that we can rewrite (6.1) as 1 b b) = argmin p 1 ——Y XB——2 + ——B——1,p s. t. (B, =p ——Y XB——F . (6.2) F nm nm B, Since in (6.2) is not specific to any individual task, it cannot calibrate the regularization. Thus it is fundamentally different from CMR. 8

2 References

[1] T.W Anderson. An introduction to multivariate statistical analysis. Wiley New York, 1958. [2] Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. The Journal of Machine Learning Research, 6(11):1817?1853, 2005. [3] J Baxter. A model of inductive bias learning. Journal of Artificial Intelligence Research, 12:149?198, 2000. [4] A. Beck and M Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. IEEE Transactions on Image Processing, 18(11):2419?2434, 2009. [5] A. Belloni, V. Chernozhukov, and L Wang. Square-root lasso: pivotal recovery of sparse signals via conic programming. Biometrika, 98(4):791?806, 2011. [6] Peter J Bickel, Yaacov Ritov, and Alexandre B Tsybakov. Simultaneous analysis of lasso and dantzig selector. The Annals of Statistics, 37(4):1705?1732, 2009. [7] L. Breiman and J.H Friedman. Predicting multivariate responses in multiple linear regression. Journal of the Royal Statistical Society: Series B, 59(1):3?54, 2002. [8] Florentina Bunea, Johannes Lederer, and Yiyuan She. The group square-root

lasso: Theoretical properties and fast algorithms. IEEE Transactions on Information Theory, 60:1313 ? 1325, 2013. [9] Iain M Johnstone. Chi-square oracle inequalities. Lecture Notes-Monograph Series, pages 399?418, 2001. [10] Michel Ledoux and Michel Talagrand. Probability in Banach Spaces: isoperimetry and processes. Springer, 2011. [11] H. Liu, M. Palatucci, and J Zhang. Blockwise coordinate descent procedures for the multi-task lasso, with applications to neural semantic basis discovery. In Proceedings of the 26th Annual International Conference on Machine Learning, pages 649?656. ACM, 2009. [12] J. Liu and J Ye. Efficient ‘1 /‘q norm regularization. Technical report, Arizona State University, 2010. [13] K. Lounici, M. Pontil, S. Van De Geer, and A.B Tsybakov. Oracle inequalities and optimal inference under group sparsity. The Annals of Statistics, 39(4):2164?2204, 2011. [14] N. Meinshausen and P. B?uhlmann. Stability selection. Journal of the Royal Statistical Society: Series B, 72(4):417?473, 2010. [15] Nicolai Meinshausen and Bin Yu. Lasso-type recovery of sparse representations for high-dimensional data. The Annals of Statistics, 37(1):246?270, 2009. [16] T.M. Mitchell, S.V. Shinkareva, A. Carlson, K.M. Chang, V.L. Malave, R.A. Mason, and M.A Just. Predicting human brain activity associated with the meanings of nouns. Science, 320(5880):1191?1195, 2008. [17] Sahand N. Negahban, Pradeep Ravikumar, Martin J. Wainwright, and Bin Yu. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. Statistical Science, 27(4):538?557, 2012. [18] Y. Nesterov. Smooth minimization of non-smooth functions. Mathematical Programming, 103(1):127? 152, 2005. [19] G. Obozinski, M.J. Wainwright, and M.I Jordan. Support union recovery in high-dimensional multivariate regression. The Annals of Statistics, 39(1):1?47, 2011. [20] Hua Ouyang, Niao He, Long Tran, and Alexander Gray. Stochastic alternating direction method of multipliers. In Proceedings of the 30th International Conference on Machine Learning, pages 80?88, 2013. [21] Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Restricted eigenvalue properties for correlated gaussian designs. The Journal of Machine Learning Research, 11(8):2241?2259, 2010. [22] A.J. Rothman, E. Levina, and J Zhu. Sparse multivariate regression with covariance estimation. Journal of Computational and Graphical Statistics, 19(4):947?962, 2010. [23] R. Tibshirani. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society, Series B, 58(1):267?288, 1996. [24] B.A. Turlach, W.N. Venables, and S.J Wright. 47(3):349?363, 2005.

Simultaneous variable selection.

Technometrics,

[25] M. Yuan and Y Lin. Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society: Series B, 68(1):49?67, 2005. [26] Jian Zhang. A probabilistic framework for multi-task learning. PhD thesis, Carnegie Mellon University, Language Technologies Institute, School of Computer Science, 2006.

9