FAST METHODS FOR RECOVERING SPARSEPARAMETERS IN LINEAR LOW RANK MODELS

Ashkan Esmaeili, Arash Amini, and Farokh Marvasti

Electrical Engineering Department, Sharif University of Technology and

Advanced Communications Research Center, Tehran, Iran

ABSTRACT

In this paper, we investigate the recovery of a sparse weight vector (parameters vector) from a set of noisy linear combinations. However, only partial information about the matrix representing the linear combinations is available. Assuming a low-rank structure for the matrix, one natural solution would be to first apply a matrix completion on the data, and then to solve the resulting compressed sensing problem. In big data applications such as massive MIMO and medical data, the matrix completion step imposes a huge computational burden. Here, we propose to reduce the computational cost of the completion task by ignoring the columns corresponding to zero elements in the sparse vector. To this end, we employ a technique to initially approximate the support of the sparse vector. We further propose to unify the partial matrix completion and sparse vector recovery into an augmented four-step problem. Simulation results reveal that the augmented approach achieves the best performance, while both proposed methods outperform the natural two-step technique with substantially less computational requirements.

BASIC PRINCIPLES

$$Y = X\beta + \epsilon \quad (1)$$

$X \in R^{m\times n}$: data matrix

$\beta \in R^{n}$: parameters signal

$\epsilon \sim N(0, I_{n\times n})$: the i.i.d noise

$Y \in R^{n}$: the observed labels.

$rank(X) = r$ where $r \ll \min(m,n)$.

$$Supp(\beta) = \{i \in \{1,\dots,n\} : \beta(i) \neq 0\}, \beta \text{ is sparse.} \quad (2)$$

$$X = \tilde{X} \odot B, where\ B_{i,j} \sim Ber(\alpha). \quad (3)$$

The Proposed Algorithm

Algorithm 1 Four-step Recovery

Input:
$\mathbf{Y_{m\times 1}}$: The vector containing the Labels
$\hat{\mathbf{X}}_{\mathbf{m\times n}}$: The data matrix containing missing entries
ϵ : Stopping criterion
$(\alpha, \lambda_1, \lambda_2)$ Algorithm Parameters
Output:
β^* : The reconstructed signal
procedure FOUR-STEP RECOVERY($\mathbf{Y}, \hat{\mathbf{X}}, \epsilon, \alpha, \lambda_1, \lambda_2$)
  Initialization: $\mathbf{X_0} \leftarrow \mathbf{0}_{m\times n}, k \leftarrow 0$
  $\beta_{\mathbf{0}_{n\times 1}} \leftarrow \mathbf{0}_{n\times 1}, \beta_{\mathbf{1}_{n\times 1}} \leftarrow \hat{\mathbf{X}}^{\dagger} \times \mathbf{Y}$
  while $||\beta_k - \beta_{k+1}||_2 > \epsilon$ **do**
    fix β_k and solve the following:
    $\mathbf{X_k} \leftarrow \min_{X_k} \ ||P_E(\mathbf{X_k} - \hat{\mathbf{X}})||_2^2 + \lambda_1||\mathbf{X_k}||_*$
    fix $\mathbf{X}_k$ and solve the following:
    $\beta^*_{k+1}(\lambda_2) \leftarrow \min_{\beta} \ ||\mathbf{X_k}\beta - Y||_2^2 + \lambda_2||\beta||_1$
    $k \leftarrow k + 1$
  end while
  $S \leftarrow \{i : \beta_i \neq 0\}$, $s \leftarrow |S|$
  Confine $\hat{\mathbf{X}}$ on supporting columns in S, and denote it with $\hat{\mathbf{X}}_{\mathbf{S}}$. Choose $\alpha << \epsilon$.
  Initialization: $\mathbf{X_0} \leftarrow \mathbf{0}_{m\times s}, k \leftarrow 0$
  $\beta_{\mathbf{0}_{n\times 1}} \leftarrow \mathbf{0}_{s\times 1}, \beta_{\mathbf{1}_{n\times 1}} \leftarrow \hat{\mathbf{X}}_{\mathbf{S}}^{\dagger} \times \mathbf{Y}$
  while $||\beta_k - \beta_{k+1}||_2 > \alpha$ **do**
    fix β_k and solve the following:
    $\mathbf{X_k} \leftarrow \min_{X_k} \ ||P_E(\mathbf{X_k} - \hat{\mathbf{X}}_{\mathbf{S}})||_2^2 + \lambda_1||\mathbf{X_k}||_*$
    fix $\mathbf{X}_k$ and solve the following:
    $\beta^*_{k+1}(\lambda_2) \leftarrow \min_{\beta} \ ||\mathbf{X_k}\beta - Y||_2^2 + \lambda_2||\beta||_1$
    $k \leftarrow k + 1$
  end while
  Correct the dimension of β by innserting 0 in other indices.
  return β^*
end procedure

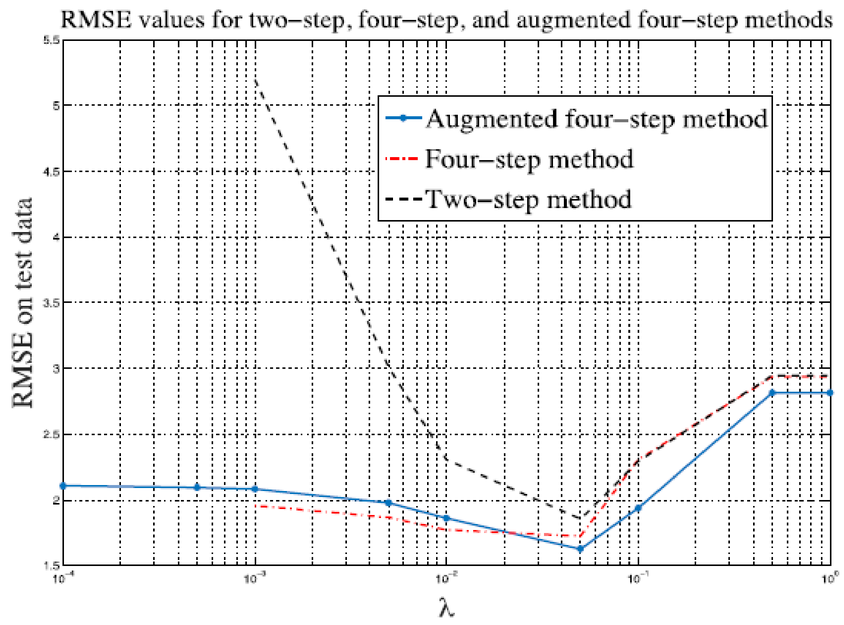
SIMULATION RESULTS

TABLE 2
Comparison between the runtimes in seconds achieved by two step, four-step, and Augmented four-step Methods

Data size \ Method	Two step method	Four step method	Augmented Four step method
m=500, n=200	0.9007	0.3333	0.5518
m=2000,n=200	1.0312	0.4674	1.0072
m=2000, n=500	8.4290	1.1630	1.1300
m=1000,n=200	1.1374	0.4020	0.6493
m=3000,n=500	8.9413	2.3973	2.5233

TABLE 3
Comparison between the RMSEs achieved by two-step, four-step, and Augmented four-step Methods

Data size \ Method	Two step method	Four step method	Augmented four step method
m=500, n=200	13.0507	18.9203	13.6528
m=2000,n=500	1.7606	1.9290	1.7555
m=2000,n=200	1.7599	1.6499	1.6648
m=1000, n=200	4.2706	3.8700	4.2629
m=3000,n=500	1.3549	1.3614	1.2998



CONCLUSION

In conclusion, we notice that large runtime is saved if we restrict the completion on the support of the initial approximation of the parameters vector without losing the performance in the prediction. In order to have an initial approximation of the parameters, we have seen that the IMAT method functions well in sparse recovery. We have found that the four-step method of initial completion followed by applying IMAT (initial sparse recovery), accurate matrix completion on reduced data, and a final sparse recovery is more efficient than the two step method of sparse recovery on the entire data followed by sparse recovery (LASSO) both in terms of the RMSE of prediction on the test set and more importantly computational efficiency. We also improved our method and called it augmented four-step method. It was observed that this method works better in terms of RMSE in comparison to the four-step method while maintaining the same (or slightly little more) amount of time complexity.