Multi-stage Multi-task feature learning via adaptive threshold

YaruFan\(^a\), YilunWang\(^{a,b}\)

\(^a\) The School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan, 611731 China
\(^b\) yilun.wang@gmail.com

Abstract

Multi-task feature learning aims to learn the shared features among tasks to improve the generalization. In [1], it was shown that minimizing non-convex optimization models, for example, one based on the capped-$\ell_1$, $\ell_1$ regularization, can obtain a better solution than the convex alternatives. In addition, an efficient multi-stage algorithm was proposed solve these non-convex models. However, they use a fixed threshold in the definition of the capped-$\ell_1$, $\ell_1$ regularization. In this paper we propose to employ an adaptive threshold in the capped-$\ell_1$, $\ell_1$ regularized formulation, and the corresponding multi-stage multi-task feature learning algorithm (MSMTFL) will incorporate a component to adaptively determine the threshold. The resulted variant of the original MSMTFL algorithm is expected to improve the ability of the original MSMTFL algorithm in terms of better feature selection performance. In particular, the adaptive threshold approach comes from iterative support detection (ISD, for short) method [2], which use the "first significant jump" rule to obtain the threshold adaptively. The rule aims to refine the threshold in each stage. Empirical studies on both synthetic and real-world data sets demonstrate the effectiveness of the new MSMTFL algorithm using an adaptive threshold approach (MSMTFL-AT, for short) over the original version.

\(^1\)Corresponding author. E-mail: yilun.wang@gmail.com.
1. Introduction

Multi-task Learning (MTL) is different from single task learning in the training (induction) process. MTL learns tasks in parallel with other related tasks at the same time, using a shared representation, and thus what is learned for each task can help other tasks be learned better. It has been successfully applied to many applications such as stock selection [4], speech classification [5] and medical diagnosis [6]. One general assumption of multi-task learning is that all tasks should share some common structures including a similarity metric matrix [4], a low rank subspace [7, 8], parameters of Bayesian models [9] and a common set of features [10, 11, 12].

In this paper, for the multi-task feature learning we focus on, there are features commonly shared among most tasks. While many multi-task feature learning algorithms have been proposed, most of them are either formulated as a convex regularization problem or assumed that the relevant features are shared by all tasks, thus the resulted models are restrictive and suboptimal. In order to remedy the shortcomings, a specific non-convex formulation based on the capped-$\ell_1, \ell_1$ regularized formulation for multi-task sparse feature learning, was proposed in [1] and this model allows a certain feature to be shared by some tasks but not all tasks. A correspondingly Multi-Stage Multi Task Feature Learning (MSMTFL, for short) algorithm is presented for the model and it usually achieve a better solution than the corresponding convex models, though it may not get a globally optimal solution.

However, this capped-$\ell_1, \ell_1$ regularized formulation is using a fixed threshold value in its definition, which is hard to estimate beforehand. In order to achieve even better performance, we propose to employ an adaptive threshold to achieve an improved capped-$\ell_1, \ell_1$ regularized model. Motivated by the iterative support detection method firstly proposed by Wang et al [2] which employed "first significant jump" rule to obtain adaptive threshold, we leverage the rule to multi-stage multi-task feature learning algorithm for solving the non-convex formulation with the adaptive threshold. With the help of adaptive threshold, an even better result can be obtained.

We will firstly review the MSMTFL algorithm in detail in following section 2. In section 3, we will present to adaptively set the threshold in MSMTFL. In section 4, several numerical experiments will verify the effectiveness of our new algorithm. In the end, the conclusion and future work will be given.
2. Review of the MSMTFL Algorithm

The MSMTFL algorithm aimed to solve the non-convex optimization model, using the concave duality [3]. Thus we first review the non-convex multi-task feature learning model as follows. We assume that \( m \) learning tasks associated with training data \( \{(X_1, y_1), \cdots, (X_m, y_m)\} \) are given, where \( X_i \in \mathbb{R}^{n_i \times d} \) is the data matrix of the \( i \)-th task with each row as a sample, \( y_i \in \mathbb{R}^{n_i} \) is the response of the \( i \)-th task, \( d \) is the data dimensionality and \( n_i \) is the number of samples for the \( i \)-th task. We learn a weight matrix \( W = [w_1, \cdots, w_m] \in \mathbb{R}^{d \times n} \), making up of the weight vectors for \( m \) linear predictors: \( y_i \approx f_i(X_i) = X_iw_i, \ i \in N_m \). The non-convex multi-task feature learning model learns these \( m \) predictors simultaneously based on the capped-\( \ell_1, \ell_1 \) regularization as well as leverages the \( \ell_1 \) penalty on each row of \( W \). Correspondingly the non-convex optimization model is as follows:

\[
\min_{W} \left\{ l(W) + \lambda \sum_{j=1}^{d} \min(\|w_j\|_1, \theta) \right\}, \tag{1}
\]

where \( \lambda(>0) \) is a parameter balancing the empirical loss and the regularization; \( \theta(>0) \) is a thresholding parameter, which is to distinguish nonzeros and zero components; \( w_j \) is the \( j \)-th row of the matrix \( W \); \( l(W) \) is the quadratic loss function of \( W \),

\[
l(W) = \sum_{i=1}^{m} \frac{1}{mn_i} \|X_iw_i - y_i\|^2. \tag{2}
\]

Obviously, due to the capped-\( \ell_1, \ell_1 \) penalty, the optimal solution of problem (1) denoted as \( W^* \) has many zero rows. Due to the \( \ell_1 \) penalty on each row of \( W \), some entries of the nonzero row \( (w^*)^k \) may be zero. Therefore, a certain feature can be shared by some tasks but not necessarily all the tasks under the formulation (1).

The formulation (1) is non-convex and thus difficult to solve. The multi-stage multi-task feature learning (MSMTFL) algorithm (see Algorithm 1) is proposed in [1] to solve (1). Note that when \( \ell = 1 \), the MSMTFL algorithm is equivalent to the \( \ell_1 \) regularized multi-task feature learning algorithm (Lasso). Therefore, the solution of MSMTFL algorithm can be considered as a refinement of that of Lasso. Although Algorithm 1 may not find a globally optimal solution, the solution may still have better performance than the solution of
Lasso. [1] has theoretically showed that the solution obtained by Algorithm 1 improves the performance of the parameter estimation error bound during the multi-stage iteration under certain circumstances. The full version [13] provides more details about intuitive interpretations, convergence analysis and reproducibility discussions of the MSMTFL algorithm.

Algorithm 1 The MSMTFL Algorithm
1. Initialize $\lambda^{0}_{j} = \lambda$;
2. for $\ell = 1, 2, \ldots$, do
   Let $\hat{W}^{(\ell)}$ be a solution of the following problem:
   $$\min_{W} \{l(W) + \sum_{j=1}^{d} \lambda^{(\ell-1)}_{j} \|w^{j}\|_{1}\}.$$ 
   Let $\lambda^{(\ell)}_{j} = \lambda I(\|\hat{w}^{(\ell)}^{j}\|_{1} < \theta)(j = 1, \cdots, d)$, where $(\hat{w}^{(\ell)}^{j})^{j}$ is the $j$-th row of $\hat{W}^{(\ell)}$ and $I(\cdot)$ denoted the $\{0, 1\}$ valued indicator function.
end.

3. Our Proposed Formulation and Algorithm

Intuitively, the formulation (1) employed a fixed threshold $\theta$ to learn a weight matrix, which is far from optimal because the choice of $\theta$ may vary a lot for different kinds of training data $X$ and hard to be determined beforehand. For the sake of achieving much better performance, we propose to employ an adaptive threshold on the non-convex multi-task feature learning formulation based on formulation (1). It is important that we can refine threshold adaptively in each stage to learn the most true nonzeros information of $W$. In particular, we makes use of the “first significant jump” rule proposed in [?] automatically learn different kinds of training data to generate an appropriate threshold value.

Here, we still the same settings and notations as in section 2. We will study a more general multi-stage framework in the multi-task setting and exploit the relationship among tasks, by using the relations of sparse feature and treating the $\ell_{1}$-norm on each row of the weight matrix as a whole for consideration with an adaptive threshold. Specifically, we propose to modified the original non-convex multi-task feature learning formulation (1) with an adaptive threshold value, as follows.

$$\min_{W}\{l(W) + \lambda \sum_{j=1}^{d} \min(t_{j}, \theta^{(\ell)})\},$$

(3)
where $l(W)$ still is the quadratic loss function (2); $\lambda(>0)$ is a parameter balancing the quadratic loss and the regularization; $t[j] = \|w^j\|_1$ is a column vector. We can see that the difference between the models (3) and (1) is the threshold parameter. In the new model, $\theta(\ell)$ is data dependent, not just a prescribe value. While it seems to be a natural idea, the adaptive determination of the $\theta(\ell)$ is not trivial and we will present an efficient method based on the "first significant jump" in Section 3.1.

To solve the above optimization problem, we correspondingly modified the multi-stage multi-task feature learning algorithm proposed in [1] by incorporating an adaptive threshold approach (MSMTFL-AT) (see Algorithm 2). At each iteration in the Algorithm 2, the threshold $\theta(\ell)$ will change according to the most recently learned solution $\hat{W}(\ell)$, which makes the MSMTFL-AT less sensitive to $\theta(\ell)$. The "first significant jump" rule will be introduced in more details in next sections and we provide a demo to demonstrate our algorithm can exactly learn the true weight matrix $\hat{W}$.

Note that the non-convex formulation (3) based on capped-$\ell_1$, $\ell_1$ regularization is a good approximation to $L_0$ because as $\theta(\ell) \rightarrow 0$, $\sum_{j=1}^d \min(t[j], \theta(\ell))/\theta(\ell) \rightarrow ||t||_0$. Therefore when $\theta(\ell) \rightarrow 0$, this formulation condition is equivalent to the sparse $L_0$ regularization up to a rescaling of $\lambda$. In practice, the $\theta(\ell)$ indeed usually gradually decreases as the iteration proceeds, though not necessarily always monotonically.

---

**Algorithm 2** The MSMTFL-AT Algorithm

1. Initialize $\lambda_0^j = \lambda$ ;
2. for $\ell = 1, 2, \cdots$, do
   a. $\hat{W}(\ell) \leftarrow \min_W \{\ell(W) + \sum_{j=1}^d \lambda_j^{(\ell-1)} t[j]\}$;
   b. $\theta(\ell) \leftarrow \text{"first significant jump" rule, using } \hat{W}(\ell) \text{ as the reference};$
   c. $\lambda_j^{(\ell)} = \lambda I(\hat{\ell}_j[j] < \theta(\ell)) \quad (j = 1, \cdots, d),$
   where $\hat{\ell}_j[j] = \|(\hat{w}(\ell))^j\|_1$ ($\hat{w}(\ell)^j$ is the $j$-th row of $\hat{W}(\ell)$)
   and $I(\cdot)$ denoted the $\{0, 1\}$ valued indicator function ;
end.

---

3.1. adaptive threshold in MSMTFL-AT algorithm

While there could be many different rules for $\theta(\ell)$, our choice is based on locating the "first significant jump" rule in the increasingly sorted sequence $|\hat{t}_j|$, where $|\hat{t}_j|$ denotes the $j$-th largest component of $\hat{t}$ by magnitude, as
used in [2]. The rule looks for the smallest \( j \) such that

\[
|\hat{t}_{[j]}^{(\ell)}| - |\hat{t}_{[j]}^{(\ell+1)}| > \tau^{(\ell)}.
\]  

This amounts to sweeping the increasing sequence \( |\hat{t}_{[j]}^{(\ell+1)}| \) and looking for the first jump larger than \( \tau^{(\ell)} \). Then we set \( \theta^{(\ell)} = |\hat{t}_{[j]}^{(\ell)}| \). Obviously, the rule has proved to be able to detect lots of true nonzeros with few false alarms if the sequence \( |\hat{t}_{[j]}^{(\ell)}| \) has the fast decaying property. Some simple and heuristic method has been adopted to define \( \tau^{(\ell)} \) for different kinds of data matrix, as suggested in [?]. For example, when each element of the data matrix \( X_i \) is sampled from the Gaussian distribution, one can set \( \tau^{(\ell)} = n^{-1} \|\hat{t}^{(\ell)}\|_\infty \), where \( n \) is the number of samples. We need to pointed out the tuning parameter \( \theta^{(\ell)} > 0 \) is a key parameter, which typically decreases from a large value to a small value to extract more correct nonzero information from those intermediate learning results as the iteration proceeds (as \( \ell \) increases).

The "first significant jump" rule works partially because in \( \hat{t}^{(\ell)} \), the true nonzeros are large in size and small in number, while the false ones are large in number and small in size. Therefore, the magnitudes of the true nonzeros are spread out, while those of the false ones are clustered. The false ones are the smearing due to the nonzeros in \( \bar{t} \) (\( \bar{t} = \|\bar{W}^j\|_1 \)) that are zeroed out in \( \hat{t}^{(\ell)} \). Some theoretical analysis are presented in [2] that if the false nonzeros are not too many, the improvement can still be expected.

3.2. A Simple Demo

We present a demo to show the effectiveness of "first significant jump" rule in multi-stage multi-task feature learning, as illustrated by Figure 1. We generated a sparse weight matrix \( \bar{W} \in R^{200 \times 20} \) and randomly set 90% rows of it as zero vectors and 80% elements of the remaining nonzero entries as zeros. We let \( n = 30 \) and create a data matrix \( X \in R^{30 \times 200} \) from the Gaussian distribution \( N(0, 1) \). The noise \( \delta \in R^{30 \times 20} \) is sampled i.i.d from the Gaussian distribution \( N(0, \sigma^2) \) with \( \sigma = 0.005 \). The responses are computed as \( Y = X\bar{W} + \delta \). The parameter estimation error is defined as \( \|\hat{W} - \bar{W}\|_{2,1} \) used in Figure 1.

In Figure 1, we use \( \bar{t} \) (a column vector) on behalf of the true weight matrix \( \bar{W} \) and \( \hat{t}^{(\ell)} \) on behalf of the learning weight matrix \( \hat{W} \) in the \( \ell \)-th iteration. Subgraph (a), (b), (c) and (d) respectively perform the learning weight matrix \( \hat{W}^{(1)} \), \( \hat{W}^{(4)} \), \( \hat{W}^{(7)} \) and \( \hat{W}^{(10)} \) obtained by our MSMTFL-AT,
in comparison with the true weight matrix $\bar{W}$. For subgraph (a), it is clear that $\hat{t}(1)$, which represented the solution of Lasso algorithm, contained a large number of false nonzeros and had a larger error. However, the $\hat{t}(4)$ show our algorithm has already started working. Notably, most of true nonzeros with large magnitude had been correctly learned. Then, $\hat{t}(7)$ well matched the true learning matrix $\bar{t}$ except for tiny number of false nonzero components. And $\hat{t}(10)$ had exactly the same nonzero components as $\bar{t}$, as well as the error was decreased rapidly. In short, Figure 1 shows our proposed algorithm is insensitive to a small false number in $\hat{t}$ and has an attractive self-correction capacity.

Figure 1: a demo of "first significant jump" rule in multi-stage multi-task feature learning. (a)1st iteration, Error=1.59e+02, (b)4th iteration, Error=6.40e+01, (c)7th iteration, Error=2.24e+01, (d)10th iteration, Error=2.10e+00.

4. Numerical Experiments

In this section, we demonstrate the better performance of the proposed MSMTFL-AT through solving the non-convex optimization problem in terms of smaller recovery errors. We compare our proposed MSMTFL-AT with
three competing multi-task feature learning algorithms: the MSMTFL algorithm, \( \ell_1 \)-norm multi-task feature learning algorithm (Lasso) and \( \ell_{2,1} \)-norm multi-task feature learning algorithm (L2,1).

4.1. Synthetic Data Experiments

Like [1], we set the number of tasks as \( m \) and each task has \( n \) samples which are of dimensionality \( d \). Each element of the data matrix \( X_i \in \mathbb{R}^{n \times d} \) for the \( i \)-th task is sampled i.i.d. from the Gaussian distribution \( N(0,1) \) and each entry of the true weight \( \bar{W} \in \mathbb{R}^{d \times m} \) is sampled i.i.d. from the uniform distribution in the interval \([-10, 10]\). Here we also randomly set 90% rows of \( \bar{W} \) as zero vectors and 80% elements of the remaining nonzero entries as zeros. Each entry of the noise \( \delta_i \in \mathbb{R}^n \) is sampled i.i.d. from the Gaussian distribution \((0, \sigma^2)\). The responses are computed as \( y_i = X_i \bar{w}_i + \delta_i \).

The quality of obtained weight matrix is measured by the averaged parameter estimation error \( \| \hat{W} - \bar{W} \|_{2,1} \) due to its having a theoretical bound referring to [1]. We present the averaged parameter estimation error \( \| \hat{W} - \bar{W} \|_{2,1} \) vs. Stage \( \ell \) compare MSMTFL algorithm and MSMTFL-AT in Figure 2. It has been seen that the error of all tested algorithms decrease with stage \( \ell \) increasing, which shows the advantage of MSMTFL and MSMTFL-AT over Lasso (\( \ell = 1 \)). Then, our proposed algorithm is always superior to MSMTFL algorithm in the same parameter settings. Moreover, the parameter estimation error of our MEMTFL-AT decreases quickly and converges in a few stages just as MSMTFL algorithm. It suggests that not only our proposed algorithm improves the ability of MSMTFL algorithm to learn weight matrix but also guarantees its convergence.

In Figure 3, we present the averaged parameter estimation error \( \| \hat{W} - \bar{W} \|_{2,1} \) vs. lambda \( \lambda \) in comparison with the MSMTFL-AT, the MSMTFL algorithm with different \( \theta \) (\( \theta_1 = 50m\lambda \), \( \theta_2 = 10m\lambda \), \( \theta_3 = 2m\lambda \) and \( \theta_4 = 0.4m\lambda \)), \( \ell_1 \)-norm multi-task feature learning algorithm (Lasso) and \( \ell_{2,1} \)-norm multi-task feature learning algorithm (L2,1). We compare the smallest averaged parameter estimation errors of all tested algorithms in all the parameter settings [15, 16]. As expected, the error of our MSMTFL-AT is the smallest among all tested algorithms. These empirical results demonstrate the effectiveness of our MSMTFL-AT. In addition, we observe that (a) when noise level (\( \sigma = 0.05 \)) is relatively large, the MSMTFL-AT still outperforms other tested algorithms. (b) when \( \lambda \) exceeds certain, the errors of all tested algorithms increase, which is reasonable due to the solutions \( \hat{W} \) obtained by the four algorithms are all too sparse.
Figure 2: Averaged parameter estimation error $\|\hat{W} - \bar{W}\|_{2,1}$ vs. Stage ($\ell$) plots for MSMTFL-AT and MSMTFL algorithms on the synthetic data set (averaged over 10 runs). We set $\lambda = \alpha \sqrt{\ln(dm)/n}$, $\theta = 50m\lambda$ referring to [1]. (a)$m=20$, $n=30$, $d=200$, $\sigma=0.005$ (b)$m=15$, $n=40$, $d=250$, $\sigma=0.01$ (c)$m=25$, $n=25$, $d=180$, $\sigma=0.05$.

Figure 3: Averaged parameter estimation error $\|\hat{W} - \bar{W}\|_{2,1}$ vs. Lambda ($\lambda$) plots for Lasso, L2,1, MSMTFL-AT and MSMTFL(theta1,2,3,4) algorithms on the synthetic data set (averaged over 10 runs). We set $\theta_1 = 50m\lambda$, $\theta_2 = 10m\lambda$, $\theta_3 = 2m\lambda$ and $\theta_4 = 0.4m\lambda$ for MSMTFL referring to [1]. (a)$m=20$, $n=30$, $d=200$, $\sigma=0.005$ (b)$m=15$, $n=40$, $d=250$, $\sigma=0.01$ (c)$m=25$, $n=25$, $d=180$, $\sigma=0.05$. 
4.2. Real-World Data Experiments

We conduct a real-world data set, i.e. the Isolet data set, to demonstrate the high efficiency of our algorithm. But it doesn’t mean our algorithm is only appropriate for this data sets. The Isolet data sets is collected from 150 speakers who speak the name of each English letter of the alphabet twice. Hence, we have 52 training samples from each speaker. The speakers are grouped into 5 subsets of 30 similar speakers each, and the subsets are referred to as isolet1, isolet2, isolet3, isolet4, and isolet5, respectively. Thus, there are 5 tasks with each task corresponding to a subset. The 5 tasks respectively have 1560, 1560, 1560, 1558 and 1559 samples (Three samples historically are missing), where each sample includes 617 features and the response is the English letter label (1-26).

For our experiments, the letter labels are treated as the regression values for the Isolet sets. We randomly extract the training samples from each task with different training ratios (15%, 20% and 25%) and use the rest of samples to form the test set. We evaluate three multi-task feature learning algorithms according to normalized mean squared error (nMSE) and averaged mean squared error (aMSE), whose definitions are as follows:

\[
nMSE = n \frac{||\hat{y} - \bar{y}||_2^2}{||\bar{y}||_1 \cdot ||\hat{y}||_1},
\]

\[
aMSE = \frac{||\hat{y} - \bar{y}||_2}{||\hat{y}||_2},
\]

where \(\hat{y}\) is the predictive value from tested algorithms and \(\bar{y}\) is the true value from Isolet data sets. Both nMSE and aMSE are commonly used in multi-task learning problems [1, 19].

The experimental results are shown in Figure 4. We see our proposed MSMTFL-AT outperforms Lasso and MSMTFL algorithm in terms of achieving the smallest nMSE and aMSE. The MSMTFL-AT performs well even in the case of a small training ratio. Our results suggest that the proposed algorithm is a better approach.

5. Conclusions

This paper proposes a non-convex multi-task feature learning formulation with an adaptive threshold and introduces the multi-stage multi-task feature
learning algorithm using an adaptive threshold approach. The MSMTFL-AT is kind of the combination of "first significant jump” rule [2] with the MSMTFL algorithm [1]. The intuition is to refine threshold iteratively by using solutions obtained from earlier stages. This leads to a more accurate threshold in each stage, and thus gradually improved solutions. The experimental results on both synthetic and real-world data demonstrate the effectiveness of our proposed MSMTFL-AT in comparison with the state of the art multi-task feature learning algorithms. In the future, we will extend our work to more non-convex or convex regularization for multi-task feature learning and to more practical applications.

Acknowledgements

This work was supported by the Natural Science Foundation of China, Grant Nos. 11201054, 91330201 and by the Fundamental Research Funds for the Central Universities ZYGX2012J118.

References

[1] P. Gong, J. Ye and C.Zhang. Multi-Stage Multi-Task Feature Learning. The Journal of Machine Learning Research archive Volume 14 Issue 1, January 2013 Issue-in-Progress Pages 2979-3010.
[2] Y. Wang and W. Yin. Sparse signal reconstruction via iteration support detection. SIAM Journal on Imaging Sciences, vol. 3, no. 3, pp. 462-491, 2010.

[3] T. Zhang. Analysis of multi-stage convex relaxation for sparse regularization. JMLR, 11: 1081C1107, 2010.

[4] J. Ghosn and Y. Bengio. Multi-task learning for stock selection. Advances in Neural Information Processing Systems, 1997

[5] S. Parameswaran and K. Weinberger. Large margin multi-task metric learning. In NIPS, pages 1867C1875, 2010.

[6] J. Bi, T. Xiong, S. Yu, M. Dundar, and R. Rao. An improved multi-task learning approach with applications in medical diagnosis. Machine Learning and Knowledge Discovery in Databases, pages 117C132, 2008.

[7] J. Chen, J. Liu, and J. Ye. Learning incoherent sparse and low-rank patterns from multiple tasks. In SIGKDD, pages 1179C1188, 2010.

[8] S. Negahban and M. Wainwright. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. The Annals of Statistics, 39(2):1069-1097, 2011.

[9] K. Yu, V. Tresp, and A. Schwaighofer. Learning gaussian processes from multiple tasks. In ICML, pages 1012C1019, 2005.

[10] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. Machine Learning, 73(3):243C272, 2008.

[11] S. Kim and E. Xing. Tree-guided group lasso for multi-task regression with structured sparsity. In ICML, pages 543C550, 2009.

[12] K. Lounici, M. Pontil, A. Tsybakov, and S. Van De Geer. Taking advantage of sparsity in multi-task learning. In COLT, pages 73C82, 2009.

[13] P. Gong, J. Ye, and C. Zhang. Multi-stage multi-task feature learning. arXiv:1210.5806, 2012.

[14] C. Zhang and T. Zhang. A general theory of concave regularization for high dimensional sparse estimation problems. Statistical Science, 2012.
[15] T. Zhang. Some sharp performance bounds for least squares regression with $\ell_1$ regularization. The Annals of Statistics, 37:2109C2144, 2009.

[16] T. Zhang. Analysis of multi-stage convex relaxation for sparse regularization. JMLR, 11:1081C1107, 2010.

[17] T. Zhang. Multi-stage convex relaxation for feature selection. Bernoulli, 2012.

[18] S. Van De Geer and P. Buhlmann. On the conditions used to prove oracle results for the lasso. Electronic Journal of Statistics, 3:1360C1392, 2009.

[19] Y. Zhang and D. Yeung. Multi-task learning using generalized t process. In AISTATS, 2010.