Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

Multi-Task Learning and Algorithmic Stability

Yu Zhang
Department of Computer Science, Hong Kong Baptist University
The Institute of Research and Continuing Education, Hong Kong Baptist University (Shenzhen)

Abstract

In this paper, we study multi-task algorithms from the perspective of the algorithmic stability. We give a definition of the multi-task uniform stability, a generalization of the conventional uniform stability, which measures the maximum difference between the loss of a multi-task algorithm trained on a data set and that of the multi-task algorithm trained on the same data set but with a data point removed in each task. In order to analyze multi-task algorithms based on multi-task uniform stability, we prove a generalized McDiarmid's inequality which assumes the difference bound condition holds by changing multiple input arguments instead of only one in the conventional McDiarmid's inequality. By using the generalized McDiarmid's inequality as a tool, we can analyze the generalization performance of general multi-task algorithms in terms of the multi-task uniform stability. Moreover, as applications, we prove generalization bounds of several representative regularized multi-task algorithms.

Introduction

Multi-task learning (Caruana 1997) aims to learn multiple related tasks simultaneously to improve the generalization performance of each task by leveraging the useful information among all tasks. Multi-task learning has many real-world applications in various areas, such as data mining, computer vision, bioinformatics, healthcare, and so on.

In the past decades, many multi-task learning models have been proposed. Among all the existing models, the multi-layer feedforward neural network (Caruana 1997), where the hidden layers provide the common feature representations for all tasks, is among the earliest learning models for multi-task learning. Then some regularized methods are developed for multi-task learning by adapting the single-task learners. One example is the multi-task support vector machine in (Evgeniou and Pontil 2004) which proposes a regularizor to enforce the model parameters of all tasks to approach the average. Some other representative regularized multi-task models include learning shared subspace (Ando and Zhang 2005), multi-task feature selection (Obozinski, Taskar, and Jordan 2006) by using the group sparse regularization, learning shared feature covariance (Argyriou, Evgeniou, and Pontil 2006), learning from the given task relations (Evgeniou, Micchelli, and Pontil 2005; Kato et al. 2007), learning the task relation from data in different ways (Jacob, Bach, and Vert 2008; Zhang and Yeung 2010a; Solnon, Arlot, and Bach 2012; Zhang and Yeung 2013; 2014), decomposition methods (Chen, Liu, and Ye 2010; Jalali et al. 2010; Lozano and Swirszcz 2012; Zweig and Weinshall 2013) which decomposes the model parameters into two or more parts with each part modeling one type of sparsity, and constructing multi-task local learners (Zhang 2013). As well as the aforementioned multi-task models, some Bayesian models have been extended to multi-task setting by using Gaussian process (Bonilla, Chai, and Williams 2007), Dirichlet process (Xue et al. 2007), t process (Zhang and Yeung 2010b) and so on.

Besides the algorithmic development, several theoretical analysis has been conducted for multi-task learning. For example, Baxter (2000) analyzes the multi-task generalization bound via the VC dimension. Also based on the VC dimension, Ben-David and Borbely (2008) study the generalization bound for multi-task algorithms where the data distributions of any pair of tasks are assumed to be transformable. Ando and Zhang (2005) provide a generalization bound for a multi-task method, which learns the shared subspace among tasks, by using the covering number. Maurer (2006) uses the Rademacher complexity to analyze linear multi-task methods. Kolar et al. (2011) investigate to recover sparse patterns. Kakade et al. (2012) investigate the regularization techniques for learning with matrices and apply the analysis to multi-task learning. The generalization bound for the multi-task sparse coding is analyzed in (Maurer, Pontil, and Romera-Paredes 2013; Maurer, Pontil, and Romera-Paredes 2014). Pontil and Maurer (2013) study the generalization bound of multi-task algorithms which upper-bound the trace norm as a constraint to learn low-rank parameter matrix. Besides the generalization bound, the oracle inequality is studied in (Lounici et al. 2009; Solnon, Arlot, and Bach 2012; Solnon 2013). Even though there is so much analysis for multi-task learning, most of them analyze multi-task algorithms whose objective functions are formulated as constrained optimization problems with the constraints achieving regularization on the model pa-

rameters. However, many algorithms with regularized instead of constrained optimization problems have been proposed for multi-task learning and hence few of the above analysis can be a principal tool to analyze those regularized algorithms. As a complementary tool to the existing ones for multi-task learning such as the VC dimension, covering number, and Rademacher complexity, the algorithmic stability (Bousquet and Elisseeff 2002) is very suitable to analyze regularized single-task learning algorithms. According to the analysis in (Bousquet and Elisseeff 2002), the algorithmic stability has a close connection to the generalization in that an algorithm with its stability coefficient satisfying some property can generalize. Even though the stability is so important, to the best of our knowledge, there is no work to even define stability for general multi-task algorithms where different tasks have different training datasets.

In this paper, we aim to fill this gap and provide a principal tool to analyze regularized multi-task algorithms. In order to achieve this objective, first we define the multi-task uniform stability which is to measure the maximum difference between the loss of a multi-task algorithm trained on a data set and that of the multi-task algorithm trained on the same data set but with one data point of each task removed. The multi-task uniform stability is different from the single-task uniform stability since m data points, where m is the number of tasks, are removed from the training set instead of only one data point removed in the single-task uniform stability. In order to analyze multi-task algorithms based on the multi-task uniform stability, we prove a generalized McDiarmid's inequality which assumes that the difference bound condition holds when replacing multiple input arguments of a function instead of only one replaced in the conventional McDiarmid's inequality. By using the generalized McDiarmid's inequality as a tool, we can analyze the generalization bound of multi-task algorithms in terms of the multi-task uniform stability. Moreover, as applications, we analyze the generalization bounds of several representative regularized multi-task algorithms including learning with task covariance matrix, learning with trace norm, and learning with composite sparse norms. To the best of our knowledge, we are not aware of any other work to analyze those regularized multi-task algorithms.

Stability of Multi-Task Algorithms

In this section, we define the stability for multi-task algorithms.

To facilitate the presentation, we first introduce some notations. Suppose we are given m learning tasks $\{T_i\}_{i=1}^m$. The training set for each task T_i consists of n_i data points $\{(\mathbf{x}_j^i, y_j^i)\}_{j=1}^{n_i}$ with the data $\mathbf{x}_j^i \in \mathbb{R}^d$ and its label $y_j^i \in \{-1, 1\}$ corresponding to a binary classification problem or $y_j^i \in \mathbb{R}$ for a regression problem. A training example $\mathbf{z}_j^i$ is defined as a pair of a data point $\mathbf{x}_j^i$ and its label y_j^i, i.e., $\mathbf{z}_j^i = (\mathbf{x}_j^i, y_j^i)$. The training set of the ith task is defined as $\mathcal{S}_i = \{\mathbf{z}_1^i, \ldots, \mathbf{z}_{n_i}^i\}$ which are drawn from some unknown distribution $\mathcal{D}_i$. $\mathcal{S}$ denotes the union of the training sets of all tasks. A multi-task algorithm, which maps the training set $\mathcal{S}$ to a function, is denoted by $\mathcal{A}_S$ where the subscript indicates the training set. Each algorithm $\mathcal{A}_S$ is assumed to be symmetric with respect to $\mathcal{S}$, i.e., it does not depend on the order of the training data points in the training set. Given $\mathcal{S}_i$, we can modify the training set by removing the jth element, i.e., $\mathcal{S}_i^{\setminus j} = \{\mathbf{z}_1^i, \ldots, \mathbf{z}_{j-1}^i, \mathbf{z}_{j+1}^i, \ldots, \mathbf{z}_{n_i}^i\}$, or replacing the jth element, i.e., $\mathcal{S}_i^j = \{\mathbf{z}_1^i, \ldots, \mathbf{z}_{j-1}^i, \hat{\mathbf{z}}_j^i, \mathbf{z}_{j+1}^i, \ldots, \mathbf{z}_{n_i}^i\}$ where the replacement $\hat{\mathbf{z}}_j^i$ is drawn from $\mathcal{D}_i$ and is independent of $\mathcal{S}_i$. Similarly, the removing and replacing operators can be defined for $\mathcal{S}$, i.e., $\mathcal{S}^{\setminus \mathcal{I}} = \{\mathcal{S}_i^{\setminus \mathcal{I}_i} | 1 \leq i \leq m\}$ and $\mathcal{S}^{\mathcal{I}} = \{\mathcal{S}_i^{\mathcal{I}_i} | 1 \leq i \leq m\}$ where $\mathcal{I}$ is a $m \times 1$ vector with its ith element $\mathcal{I}_i$ ($1 \leq \mathcal{I}_i \leq n_i$) as the index of the data point to be removed or replaced in the ith task. The loss of a hypothesis f with respect to an example $\mathbf{z} = (\mathbf{x}, y)$ is defined as $l(f, \mathbf{z}) = c(f(\mathbf{x}), y)$ where $c(\cdot, \cdot)$ is a cost function. The generalization error of a multi-task algorithm $\mathcal{A}$ which is trained on a training set $\mathcal{S}$ is defined as

$$R(\mathcal{A}, \mathcal{S}) = \sum_{i=1}^{m} \mathbb{E}_{\mathbf{z}_i \sim \mathcal{D}_i}[l(\mathcal{A}_S, \mathbf{z}_i)], \tag{1}$$

where $\mathbb{E}[\,\cdot\,]$ denotes the expectation operator and $\mathbf{z} \sim \mathcal{D}_i$ means that $\mathbf{z}$ is sampled from distribution $\mathcal{D}_i$. The empirical loss of a multi-task algorithm $\mathcal{A}$ trained on a training dataset $\mathcal{S}$ is defined as

$$R_{emp}(\mathcal{A}, \mathcal{S}) = \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} l(\mathcal{A}_S, \mathbf{z}_j^i). \tag{2}$$

There are many types of the stability in (Bousquet and Elisseeff 2002). In this paper, our study mainly focuses on the uniform stability and gives a definition for the uniform stability under the multi-task setting in the following. For other types of stability, we can generalize them to the multi-task setting in a similar way and this is left for our future investigation.

Definition 1 (*Multi-Task Uniform Stability*) *A multi-task algorithm $\mathcal{A}$ has uniform stability τ with respect to the loss function l if the following holds*

$$\forall \mathcal{S},\ \forall \mathcal{I},\ \forall \mathbf{z}_i \sim \mathcal{D}_i,\ \left| \sum_{i=1}^{m} \Big(l(\mathcal{A}_S, \mathbf{z}_i) - l(\mathcal{A}_{S^{\setminus \mathcal{I}}}, \mathbf{z}_i) \Big) \right| \leq \tau.$$

And τ is called the stability coefficient of algorithm $\mathcal{A}$.

Remark 1 *Note that the multi-task uniform stability is different from the single-task uniform stability in (Bousquet and Elisseeff 2002). The single-task uniform stability compares the losses between a single-task algorithm trained on a dataset and the single-task algorithm trained on the same dataset but with only one training data point removed. However, the multi-task uniform stability compares the loss of a multi-task algorithm trained on a dataset with that of the multi-task algorithm trained on the same dataset by removing m training data points with one for each task. Moreover, the multi-task uniform stability can be viewed as a generalization of the single-task uniform stability since when $m = 1$ the multi-task uniform stability reduces to the single-task uniform stability.*

Then we define a σ-admissible loss function which is useful for the subsequent analysis.

Definition 2 *A loss function $l(\cdot,\cdot)$ is σ-admissible if the associated cost function $c(\cdot,\cdot)$ is convex with respect to its first argument and the following condition holds*

$$\forall y_1,\ y_2,\ y_3,\ |c(y_1,y_3)-c(y_2,y_3)| \leq \sigma|y_1-y_2|.$$

Recall that the main tool to study the single-task generalization bound based on the single-task uniform stability in (Bousquet and Elisseeff 2002) is the McDiarmid's inequality (McDiarmid 1989). However, in the definition of the multi-task uniform stability, there are m data points removed in the training set instead of only one data point removed in the definition of the single-task uniform stability and so the McDiarmid's inequality cannot be applied here. In order to study the multi-task generalization bound based on the multi-task uniform stability, in the following we prove a generalization of the McDiarmid's inequality which can allow more than one input argument of the function under investigation to be changed.[1]

Theorem 1 (Generalized McDiarmid's Inequality) *Let $X_1,\ldots,X_n$ be n independent random variables taking values from some set $\mathcal{C}$, and assume that $f:\mathcal{C}^n\to\mathbb{R}$ satisfies the following bounded differences condition when changing any q input arguments:*

$$\sup_{x_1,\ldots,x_m,\hat{x}_{j_1},\ldots,\hat{x}_{j_q}} \big|f(x_1,\ldots,x_{j_1},\ldots,x_{j_q},\ldots,x_n) - f(x_1,\ldots,\hat{x}_{j_1},\ldots,\hat{x}_{j_q},\ldots,x_n)\big| \leq a,$$

where $\{j_1,\ldots,j_q\}\subset\{1,\ldots,n\}$, n is assumed to be divisible by q, and a is a fixed constant. Then for any $\varepsilon>0$, we have

$$p\Big(f(X_1,\ldots,X_n)-\mathbb{E}[f(X_1,\ldots,X_n)]\geq\varepsilon\Big) \leq \exp\left\{-\frac{2q\varepsilon^2}{na^2}\right\},$$

Remark 2 *When $q=1$, the generalized McDiarmid's inequality becomes the conventional McDiarmid's inequality with the same upper bound a and so the inequality introduced in Theorem 1 is a generalization of the conventional McDiarmid's inequality. The constant a can be generalized to multiple constants, which depend on the indices of the changed input arguments, with small modification in the proof and here for notational simplicity we just use the same upper bound.*

Generalization Bound for General Multi-Task Algorithms

In this section, we present a generalization bound for general multi-task algorithms based on the multi-task uniform stability.

Here we assume that the numbers of training data points in different tasks are the same, i.e., $n_i=n_0$ for $i=1,\ldots,m$. Based on the generalized McDiarmid's inequality introduced in Theorem 1, we can prove the following generalization bound of general multi-task algorithms by using the multi-task uniform stability.

Theorem 2 *Let $\mathcal{A}$ be a multi-task algorithm with uniform stability τ with respect to a loss function $l(\cdot,\cdot)$ such that $0\leq l(\mathcal{A}_S,\mathbf{z})\leq M$ for all $\mathbf{z}$ and $\mathcal{S}$. Then, for any $n_0\geq 1$ and any $\delta\in(0,1)$ the following bound holds with probability at least $1-\delta$,*

$$R(\mathcal{A},\mathcal{S}) \leq R_{emp}(\mathcal{A},\mathcal{S}) + 2\tau + (4n_0\tau+mM)\sqrt{\frac{\ln(1/\delta)}{2n_0}}. \tag{3}$$

Remark 3 *Note that Theorem 2 requires that only the loss of the optimal learner produced by a multi-task algorithm but not for any learning function is bounded, which is easily satisfied by many regularized multi-task algorithms as we will see later.*

Remark 4 *We say that a learning algorithm generalizes if the empirical error converges in probability to the expected error when the size of the training dataset increases. In order to achieve the generalization, τ needs to satisfy that $\tau = o(n_0^{-\frac{1}{2}})$ based on Theorem 2, that is,*

$$\lim_{n_0\to+\infty}\frac{\tau}{n_0^{-\frac{1}{2}}}=0. \tag{4}$$

Based on Theorem 2, we can compare the bounds of both the single-task and multi-task learners. Suppose that the single-task and multi-task learners are of the same type, i.e., having the same loss function and similar regularizers. According to (Bousquet and Elisseeff 2002), the following single-task generalization bound for the ith task holds with probability at least $1-\delta$,

$$R_i^{ST} \leq R_{emp,i}^{ST} + 2\tau_i + (4n_0\tau_i+M)\sqrt{\frac{\ln(1/\delta)}{2n_0}}, \tag{5}$$

where R_i^{ST} and $R_{emp,i}^{ST}$ denote the single-task generalization error and the single-task empirical loss for the ith task respectively and τ_i is the stability coefficient of the single-task algorithm for the ith task. We can prove that the following bound holds with probability at least $1-\delta$

$$R^{ST} \leq R_{emp}^{ST} + 2\sum_{i=1}^{m}\tau_i + (4n_0\sum_{i=1}^{m}\tau_i + mM)\sqrt{\frac{\ln(m/\delta)}{2n_0}}, \tag{6}$$

where $R^{ST}=\sum_{i=1}^m R_i^{ST}$ and $R_{emp}^{ST}=\sum_{i=1}^m R_{emp,i}^{ST}$ are the aggregated generalization error and aggregated empirical loss of all the single-task learners on the m tasks respectively. By comparing the two bounds in Eqs. (6) and (3), we can see that those two bounds have similar structures. The first terms in the right-hand side of the two bounds are the empirical errors, the second ones are about the stability coefficients, and the last ones are to measure the confidences. For the empirical errors in the two bounds, we can assume the multi-task empirical loss is comparable with and even lower than the single-task one since multi-task algorithms can access

[1] Due to the page limit, we omit the proofs of all the theorems.

more labeled training data (i.e., the training data in all the tasks) than its single-task counterparts, making the multi-task algorithms have more expressive power. The confidence part in the multi-task bound is better than that in the aggregated single-task bound (i.e., Eq. (6)) given that τ is comparable to or smaller than $\sum_{i=1}^{m} \tau_i$ since in the aggregated single-task bound there is an additional constant m appearing in the numerator of the logarithm function. So if the multi-task uniform stability coefficient is smaller than the sum of all the single-task uniform stability coefficients, the multi-task bound is more likely to be better than the single-task bound, implying that multi-task learning is helpful in improving the performance of all tasks. This rule (i.e., $\tau \leq \sum_{i=1}^{m} \tau_i$) may be used to determine when we can use the multi-task algorithm instead of its single-task counterpart, which is an advantage of the multi-task stability over other analysis tools.

Applications to Regularized Multi-Task Algorithms

In this section, we apply the analysis in the previous section to analyze several representative regularized multi-task algorithms.

Here we consider a linear model for the ith task with the linear function defined as $f_i(\mathbf{x}) = \mathbf{w}_i^T \mathbf{x}$. The additional offset b_i is absorbed into $\mathbf{w}_i$ and correspondingly a constant 1 is included in the feature representation of each data point. Moreover, the dot product $\langle \mathbf{x}, \mathbf{x} \rangle$ for any $\mathbf{x}$ is assumed to be upper-bounded by κ^2.

The general objective function of regularized multi-task algorithms is formulated as

$$\min_{\mathbf{W}} \frac{1}{n_0} \sum_{i=1}^{m} \sum_{j=1}^{n_0} c\left(\mathbf{w}_i^T \mathbf{x}_j^i, y_j^i\right) + g(\mathbf{W}), \tag{7}$$

where $\mathbf{W} = (\mathbf{w}_1, \ldots, \mathbf{w}_m)$. The first term in problem (7) is to measure the empirical loss on the training set and $g(\mathbf{W})$, the regularization term, is to specify the relations among tasks as well as penalizing the complexity of $\mathbf{W}$. Here we assume that the loss function $l(\cdot, \cdot)$ or equivalently the cost function $c(\cdot, \cdot)$ satisfies the following two properties:

(i) there exists a constant $\eta > 0$ such that for all y, we have $c(0, y) \leq \eta$;

(ii) the loss function $l(\cdot, \cdot)$ is nonnegative and σ-admissible.

Remark 5 *Note that the two properties listed above are reasonable and are satisfied by most commonly used loss functions. For example, the hinge loss $c(y_1, y_2) = \max(0, 1 - y_1 y_2)$ satisfies the two properties and so does the square loss $c(y_1, y_2) = (y_1 - y_2)^2$ if the output space is bounded.*

To ensure that the minimizer of problem (7) exists, $g(\mathbf{W})$ is assumed to be coercive, that is, $\lim_{\|\mathbf{W}\|_F \to +\infty} g(\mathbf{W}) = +\infty$.

In the following sections, we investigate three types of regularized multi-task algorithms with different regularizers, i.e., a Frobenius-norm-style regularizer parameterized by a task covariance matrix, the (squared) trace norm regularization, and composite sparse regularizers where the model parameter consists of two components with each one capturing one type of sparsity.

Learning with Task Covariance Matrix

Here we investigate a regularized multi-task algorithm with the objective function formulated as

$$\min_{\mathbf{W}} \frac{1}{n_0} \sum_{i=1}^{m} \sum_{j=1}^{n_0} c\left(\mathbf{w}_i^T \mathbf{x}_j^i, y_j^i\right) + \mathrm{tr}(\mathbf{W} \boldsymbol{\Omega}^{-1} \mathbf{W}^T), \tag{8}$$

where $\mathrm{tr}(\cdot)$ denotes the trace of a square matrix. $\boldsymbol{\Omega}$, a positive definite matrix, can be viewed as a task covariance matrix to describe the pairwise relations between tasks. From the perspective of probabilistic modeling, the regularizer in problem (8) corresponds to a matrix-variate normal prior on $\mathbf{W}$ with $\boldsymbol{\Omega}$ describing the covariance between columns (Zhang and Yeung 2010a). Moreover, according to the analysis in (Zhang and Yeung 2010b), we can see that problem (8) is related to the weight-space view of a multi-task Gaussian process model proposed in (Bonilla, Chai, and Williams 2007). Problem (8) can be viewed as a generalization of the objective functions of regularized single-task learners. For example, when $\boldsymbol{\Omega}$ is a diagonal matrix, problem (8) will be decomposed into m independent single-task formulations with squared ℓ_2 regularization and when $m = 1$, $\boldsymbol{\Omega}$ reduces to a positive scalar, making problem (8) an ℓ_2-regularized single-task algorithm such as the ridge regression model. On the other hand, problem (8) has been studied by many existing works, such as (Evgeniou and Pontil 2004; Evgeniou, Micchelli, and Pontil 2005; Kato et al. 2007) which assume that $\boldsymbol{\Omega}$ is given as a priori information and (Jacob, Bach, and Vert 2008; Zhang and Yeung 2010a; Solnon, Arlot, and Bach 2012; Solnon 2013) which aim to learn $\boldsymbol{\Omega}$ from data in different ways.

Then we will analyze the multi-task uniform stability coefficient of the multi-task algorithm in problem (8).

Theorem 3 *The learning algorithm $\mathcal{A}$ defined by problem (8) has uniform stability coefficient τ as*

$$\tau \leq \frac{\lambda_1(\boldsymbol{\Omega}) \kappa^2 \sigma^2 m}{2 n_0},$$

where $\lambda_i(\boldsymbol{\Omega})$ is the ith largest eigenvalue of $\boldsymbol{\Omega}$.

In order to use Theorem 2 to give the generalization bound, we prove that the optimal solution of problem (8) produces bounded loss for any data point in the following theorem.

Theorem 4 *For the optimal solution $\mathbf{W}^\star$ of problem (8), we have*

$$c\left((\mathbf{w}_i^\star)^T \mathbf{x}, y\right) \leq \sigma \kappa \sqrt{\lambda_1(\boldsymbol{\Omega}) m \eta} + \eta$$

for any $\mathbf{x}$ and y, where $\mathbf{w}_i^\star$ is the ith column of $\mathbf{W}^\star$.

By combining Theorems 4 and 3 with Theorem 2, we can easily obtain the generalization bound of the multi-task algorithm in problem (8).

Remark 6 *According to Theorem 3, the stability coefficient τ of problem (8) satisfies Eq. (4), which implies that the multi-task algorithm induced by problem (8) generalizes, given that $\boldsymbol{\Omega}$ is positive definite.*

Remark 7 *According to (Bousquet and Elisseeff 2002), the single-task uniform stability coefficient of an ℓ_2-regularized*

single-task method such as the ridge regression model is upper-bounded, i.e., $\tau_i \leq \frac{\mu_i \sigma^2 \kappa^2}{2n_0}$ where $\frac{1}{\mu_i}\|\cdot\|_2^2$ is used as a regularizer with $\|\cdot\|_2$ denoting the ℓ_2 norm of a vector. For the multi-task algorithm in problem (8), its stability coefficient τ satisfies $\tau \leq \frac{\lambda_1(\mathbf{\Omega})\sigma^2 m \kappa^2}{2n_0}$ based on Theorem 3. If $\lambda_1(\mathbf{\Omega})$ is smaller than $\frac{1}{m}\sum_{i=1}^m \mu_i$, τ is likely to be smaller than $\sum_{i=1}^m \tau_i$ and hence the multi-task algorithm is likely to have smaller generalization error than the single-task algorithm.

Remark 8 *As revealed in Theorem 3, the bound on τ suggests that if we want to learn the task covariance matrix $\mathbf{\Omega}$ from data, the spectrum of $\mathbf{\Omega}$ can be used to define the regularizer. For example, we can use $\mathrm{tr}(\mathbf{\Omega})$ as a regularizer, leading to the following objective function as*

$$\min_{\mathbf{W},\mathbf{\Omega}} \frac{1}{n_0}\sum_{i=1}^m\sum_{j=1}^{n_0} c\left(\mathbf{w}_i^T\mathbf{x}_j^i, y_j^i\right) + \mathrm{tr}(\mathbf{W}\mathbf{\Omega}^{-1}\mathbf{W}^T) + \alpha\mathrm{tr}(\mathbf{\Omega}), \tag{9}$$

where α is a regularization parameter. By setting the derivative with respect to $\mathbf{\Omega}$ to be zero, we can get the analytical solution of $\mathbf{\Omega}$ as $\mathbf{\Omega}^\star = \frac{1}{\sqrt{\alpha}}(\mathbf{W}^T\mathbf{W})^{1/2}$. By plugging $\mathbf{\Omega}^\star$ into problem (9), we can find that the regularizer on $\mathbf{W}$ is just the trace norm of $\mathbf{W}$, which is what we want to analyze in the next section.

Learning with Trace Norm

The trace norm is widely used in multi-task learning as a regularizer (Argyriou, Evgeniou, and Pontil 2006; Zhang and Yeung 2010a; Pong et al. 2010; Chen, Liu, and Ye 2010; Chen, Zhou, and Ye 2011; Chen, Liu, and Ye 2012) since the trace norm regularization can learn a low-rank matrix which matches the assumption of multi-task learning that multiple task are related in terms of the model parameters. The objective function for learning with trace norm is formulated as

$$\min_{\mathbf{W}} \frac{1}{n_0}\sum_{i=1}^m\sum_{j=1}^{n_0} c\left(\mathbf{w}_i^T\mathbf{x}_j^i, y_j^i\right) + \alpha\|\mathbf{W}\|_* + \frac{\beta}{2}\|\mathbf{W}\|_F^2, \tag{10}$$

where $\|\cdot\|_*$ denotes the trace norm or equivalently the nuclear norm of a matrix, $\|\cdot\|_F$ denotes the matrix Frobenius norm, and α and β are the regularization parameters. In the following, we analyze the stability coefficient of problem (10) and prove the boundedness of the cost function at the optimal solution of problem (10).

Theorem 5 *The learning algorithm $\mathcal{A}$ defined by problem (10) has uniform stability coefficient τ as*

$$\tau \leq \frac{2\kappa^2\sigma^2 m}{\beta n_0}.$$

Theorem 6 *For the optimal solution $\mathbf{W}^\star$ of problem (10), we have*

$$c\left((\mathbf{w}_i^\star)^T\mathbf{x}, y\right) \leq \frac{\sigma\kappa}{\beta}\left(\sqrt{\alpha^2 + 2m\eta\beta} - \alpha\right) + \eta$$

for any $\mathbf{x}$ and y, where $\mathbf{w}_i^\star$ is the ith column of $\mathbf{W}^\star$.

Combining Theorem 5, 6 and 2 yields the generalization bound of problem (10).

Remark 9 *If only trace norm is used for regularization, i.e., β equals 0 in problem (10), we can see that according to Theorem 5 the corresponding stability coefficient cannot satisfy Eq. (4), and hence the corresponding multi-task algorithm, which is to learn the low-rank parameter matrix, is not stable, implying that it may not generalize. This statement is similar to a fact that sparse learning algorithms are not stable as proved in (Xu, Caramanis, and Mannor 2012). On the other hand, by adding a squared ℓ_2 regularizer in terms of the matrix Frobenius norm, the multi-task algorithm induced by problem (10) can generalize since its stability coefficient satisfies Eq. (4). Similar phenomena is also observed in single-task learning (Xu, Caramanis, and Mannor 2012).*

The squared trace norm regularization is also used for multi-task learning, e.g., (Argyriou, Evgeniou, and Pontil 2006; Zhang and Yeung 2010a). The objective function is formulated as

$$\min_{\mathbf{W}} \frac{1}{n_0}\sum_{i=1}^m\sum_{j=1}^{n_0} c\left(\mathbf{w}_i^T\mathbf{x}_j^i, y_j^i\right) + \frac{\alpha}{2}\|\mathbf{W}\|_*^2 + \frac{\beta}{2}\|\mathbf{W}\|_F^2, \tag{11}$$

Similar to the trace norm regularization, we can analyze problem (11) with the results in the following theorems.

Theorem 7 *The learning algorithm $\mathcal{A}$ defined by problem (11) has uniform stability coefficient τ as*

$$\tau \leq \frac{2\kappa^2\sigma^2 m}{(\alpha+\beta)n_0}.$$

Theorem 8 *For the optimal solution $\mathbf{W}^\star$ of problem (11), we have*

$$c\left((\mathbf{w}_i^\star)^T\mathbf{x}, y\right) \leq \sigma\kappa\sqrt{\frac{2m\eta}{\alpha+\beta}} + \eta$$

for any $\mathbf{x}$ and y, where $\mathbf{w}_i^\star$ is the ith column of $\mathbf{W}^\star$.

Remark 10 *According to Theorem 7, we can see that if only the squared trace norm is used for regularization, i.e., β being 0 in problem (11), the stability coefficient of the corresponding multi-task algorithm also satisfies Eq. (4), meaning that it is stable and also generalizes, which is different from the situation that only the trace norm is used as the regularizer.*

Remark 11 *Similar to the trace norm regularization, the squared ℓ_2 regularizer (i.e., $\frac{\beta}{2}\|\mathbf{W}\|_F^2$) can make the stability coefficient smaller and as a consequence, the generalization performance can become better.*

Remark 12 *In (Pontil and Maurer 2013), the effect of the trace norm on multi-task algorithms has been studied. However, different from multi-task algorithms investigated in problems (10) and (11), the multi-task algorithms in (Pontil and Maurer 2013) use the trace norm of the parameter matrix of all tasks to define a constraint to achieve regularization. The analysis in (Pontil and Maurer 2013) uses the Rademacher complexity as a tool and the proof is more complicated than ours. Moreover, (Pontil and Maurer 2013) only analyzes the trace norm case and it seems that the proof cannot be easily extended to analyze the squared trace norm. Similar to (Pontil and Maurer 2013), the analysis in (Kakade, Shalev-Shwartz, and Tewari 2012) can only be applied to the multi-task algorithms with the trace norm as a constraint instead of as a regularizer.*

Learning with Composite Sparse Norms

There are some works (e.g., (Chen, Liu, and Ye 2010; Jalali et al. 2010; Chen, Zhou, and Ye 2011; Chen, Liu, and Ye 2012; Gong, Ye, and Zhang 2012)) to decompose the model parameters of all tasks into two components which are assumed to pursue some sparse properties. For example, in (Chen, Liu, and Ye 2010; Chen, Zhou, and Ye 2011; Chen, Liu, and Ye 2012) one component is to capture the low-rank property of the model parameters and another one learns the (group) sparse parameters but in (Jalali et al. 2010; Gong, Ye, and Zhang 2012) both components learn the (group) sparse parameters. As an example, we investigate the method introduced in (Chen, Liu, and Ye 2010; 2012) with the objective function formulated as

$$\min_{\mathbf{L},\mathbf{S}} \quad \frac{1}{n_0}\sum_{i=1}^{m}\sum_{j=1}^{n_0} c\left((\mathbf{l}_i+\mathbf{s}_i)^T\mathbf{x}_j^i, y_j^i\right) + \alpha\|\mathbf{L}\|_* + \beta\|\mathbf{S}\|_1 + \frac{\gamma}{2}\left(\|\mathbf{L}\|_F^2 + \|\mathbf{S}\|_F^2\right), \tag{12}$$

where $\mathbf{l}_i$ and $\mathbf{s}_i$ denote the ith columns of $\mathbf{L}$ and $\mathbf{S}$ respectively and $\|\cdot\|_1$ denotes the ℓ_1 norm of a matrix, the sum of the absolute values of all the elements in the matrix. Problem (12) is slightly different from the original objective function in (Chen, Liu, and Ye 2010; 2012) by adding squared Frobenius norm as a regularizer, which is inspired by the analysis on the trace norm regularization in the previous section. Similar to the previous cases, we can analyze problem (12) with the results shown as follows.

Theorem 9 *The learning algorithm $\mathcal{A}$ defined by problem (12) has uniform stability coefficient τ as*

$$\tau \leq \frac{2\sqrt{2}\kappa^2\sigma^2 m\sqrt{m}}{\gamma n_0}.$$

Theorem 10 *For the optimal solution $\mathbf{L}^\star$ and $\mathbf{S}^\star$ of problem (12), we have*

$$c\left((\mathbf{w}_i^\star)^T\mathbf{x}, y\right) \leq \frac{2\sigma\kappa}{\gamma}\left(\sqrt{\theta^2+m\eta\gamma}-\theta\right)+\eta$$

for any $\mathbf{x}$ and y, where $\mathbf{w}_i^\star$ is the ith column of $\mathbf{L}^\star+\mathbf{S}^\star$ and $\theta=\min\{\alpha,\beta\}$.

By combining Theorems 10, 2, and 9, we can analyze the generalization bound of problem (12).

Remark 13 *Similar to the trace norm regularization, without the squared Frobenius regularizer, the multi-task algorithm corresponding to $\gamma=0$ may not generalize since its stability coefficient does not satisfy Eq. (4).*

Remark 14 *For other algorithms in (Jalali et al. 2010; Chen, Zhou, and Ye 2011; Gong, Ye, and Zhang 2012), we can analyze them in a similar way and due to page limit, we omit the results.*

Related Works

The most related work to ours is (Audiffren and Kadri 2013) which also investigates the stability of multi-task regression algorithms. However, different from our work, in (Audiffren and Kadri 2013) all the tasks share the same training dataset, making the analysis not applicable to the general multi-task setting where different tasks can have different training datasets. The uniform stability studied in (Audiffren and Kadri 2013) is just the single-task uniform stability proposed in (Bousquet and Elisseeff 2002) and so it is different from our proposed multi-task uniform stability. Moreover, stability has been applied in (Maurer 2005; Kuzborskij and Orabona 2013) to analyze transfer learning algorithms which are related to multi-task learning but have different settings since transfer learning aims to improve the performance of only the target task by leveraging the information from the source tasks. Similar to (Audiffren and Kadri 2013), the stability studied in (Maurer 2005; Kuzborskij and Orabona 2013) is still the single-task uniform stability in (Bousquet and Elisseeff 2002) but different from ours. In summary, our work is totally different from those existing works.

Conclusion

In this paper, we extend the uniform stability to the multi-task setting and use the proposed multi-task uniform stability to analyze several types of regularized multi-task algorithms by using the proposed generalized McDiarmid's inequality as a tool.

In our future work, we are interested in extending the analysis to other multi-task learning models. Moreover, we will study the multi-task extension of other stabilities defined in (Bousquet and Elisseeff 2002) based on the proposed generalized McDiarmid's inequality.

Acknowledgment

This work is supported by NSFC 61305071 and HKBU FRG2/13-14/039.

References

Ando, R. K., and Zhang, T. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research* 6:1817–1853.

Argyriou, A.; Evgeniou, T.; and Pontil, M. 2006. Multi-task feature learning. In *Advances in Neural Information Processing Systems 19*, 41–48.

Audiffren, J., and Kadri, H. 2013. Stability of multi-task kernel regression algorithms. In *Proceedings of Asian Conference on Machine Learning*, 1–16.

Baxter, J. 2000. A model of inductive bias learning. *Journal of Artifical Intelligence Research* 12:149–198.

Ben-David, S., and Borbely, R. S. 2008. A notion of task relatedness yielding provable multiple-task learning guarantees. *Machine Learning* 73(3):273–287.

Bonilla, E.; Chai, K. M. A.; and Williams, C. 2007. Multi-task Gaussian process prediction. In *Advances in Neural Information Processing Systems 20*, 153–160.

Bousquet, O., and Elisseeff, A. 2002. Stability and generalization. *Journal of Machine Learning Research* 2:499–526.

Caruana, R. 1997. Multitask learning. *Machine Learning* 28(1):41–75.

Chen, J.; Liu, J.; and Ye, J. 2010. Learning incoherent sparse and low-rank patterns from multiple tasks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1179–1188.

Chen, J.; Liu, J.; and Ye, J. 2012. Learning incoherent sparse and low-rank patterns from multiple tasks. *ACM Transactions on Knowledge Discovery from Data* 5(4):Article 22.

Chen, J.; Zhou, J.; and Ye, J. 2011. Integrating low-rank and group-sparse structures for robust multi-task learning. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 42–50.

Evgeniou, T., and Pontil, M. 2004. Regularized multi-task learning. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 109–117.

Evgeniou, T.; Micchelli, C. A.; and Pontil, M. 2005. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research* 6:615–637.

Gong, P.; Ye, J.; and Zhang, C. 2012. Robust multi-task feature learning. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 895–903.

Jacob, L.; Bach, F.; and Vert, J.-P. 2008. Clustered multi-task learning: a convex formulation. In *Advances in Neural Information Processing Systems 21*, 745–752.

Jalali, A.; Ravikumar, P. D.; Sanghavi, S.; and Ruan, C. 2010. A dirty model for multi-task learning. In *Advances in Neural Information Processing Systems 23*, 964–972.

Kakade, S. M.; Shalev-Shwartz, S.; and Tewari, A. 2012. Regularization techniques for learning with matrices. *Journal of Machine Learning Research* 13:1865–1890.

Kato, T.; Kashima, H.; Sugiyama, M.; and Asai, K. 2007. Multi-task learning via conic programming. In *Advances in Neural Information Processing Systems 20*, 737–744.

Kolar, M.; Lafferty, J. D.; and Wasserman, L. A. 2011. Union support recovery in multi-task learning. *Journal of Machine Learning Research* 12:2415–2435.

Kuzborskij, I., and Orabona, F. 2013. Stability and hypothesis transfer learning. In *Proceedings of the 30th International Conference on Machine Learning*, 942–950.

Lounici, K.; Pontil, M.; Tsybakov, A. B.; and van de Geer, S. A. 2009. Taking advantage of sparsity in multi-task learning. In *Proceedings of the 22nd Conference on Learning Theory*.

Lozano, A. C., and Swirszcz, G. 2012. Multi-level lasso for sparse multi-task regression. In *Proceedings of the 29th International Conference on Machine Learning*.

Maurer, A.; Pontil, M.; and Romera-Paredes, B. 2013. Sparse coding for multitask and transfer learning. In *Proceedings of the 30th International Conference on Machine Learning*, 343–351.

Maurer, A.; Pontil, M.; and Romera-Paredes, B. 2014. An inequality with applications to structured sparsity and multitask dictionary learning. In *Proceedings of The 27th Conference on Learning Theory*, 440–460.

Maurer, A. 2005. Algorithmic stability and meta-learning. *Journal of Machine Learning Research* 6:967–994.

Maurer, A. 2006. Bounds for linear multi-task learning. *Journal of Machine Learning Research* 7:117–139.

McDiarmid, C. 1989. On the method of bounded differences. *Surveys in combinatorics* 141(1):148–188.

Obozinski, G.; Taskar, B.; and Jordan, M. 2006. Multi-task feature selection. Technical report, Department of Statistics, University of California, Berkeley.

Pong, T. K.; Tseng, P.; Ji, S.; and Ye, J. 2010. Trace norm regularization: Reformulations, algorithms, and multi-task learning. *SIAM Journal on Optimization* 20(6):3465–3489.

Pontil, M., and Maurer, A. 2013. Excess risk bounds for multitask learning with trace norm regularization. In *Proceedings of the 26th Annual Conference on Learning Theory*, 55–76.

Solnon, M.; Arlot, S.; and Bach, F. 2012. Multi-task regression using minimal penalties. *Journal of Machine Learning Research* 13:2773–2812.

Solnon, M. 2013. Comparison between multi-task and single-task oracle risks in kernel ridge regression. *CoRR* abs/1307.5286.

Xu, H.; Caramanis, C.; and Mannor, S. 2012. Sparse algorithms are not stable: A no-free-lunch theorem. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34(1):187–193.

Xue, Y.; Liao, X.; Carin, L.; and Krishnapuram, B. 2007. Multi-task learning for classification with Dirichlet process priors. *Journal of Machine Learning Research* 8:35–63.

Zhang, Y., and Yeung, D.-Y. 2010a. A convex formulation for learning task relationships in multi-task learning. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, 733–742.

Zhang, Y., and Yeung, D.-Y. 2010b. Multi-task learning using generalized t process. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 964–971.

Zhang, Y., and Yeung, D.-Y. 2013. Learning high-order task relationships in multi-task learning. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, 1917–1923.

Zhang, Y., and Yeung, D.-Y. 2014. A regularization approach to learning task relationships in multitask learning. *ACM Transactions on Knowledge Discovery from Data* 8(3):article 12.

Zhang, Y. 2013. Heterogeneous-neighborhood-based multi-task local learning algorithms. In *Advances in Neural Information Processing Systems 26*.

Zweig, A., and Weinshall, D. 2013. Hierarchical regularization cascade for joint learning. In *Proceedings of the 30th International Conference on Machine Learning*, 37–45.