Neurocomputing 177 (2016) 242–256

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Multiple task learning with flexible structure regularization

Jian Pu [a,*], Jun Wang [b,c], Yu-Gang Jiang [d], Xiangyang Xue [d]

[a] *Institute of Neuroscience, Chinese Academy of Sciences, Shanghai, China*
[b] *School of Computer Science and Software Engineering, East China Normal University, Shanghai, China*
[c] *Institute of Data Science and Technology, Alibaba Group, Seattle, USA*
[d] *School of Computer Science, Fudan University, Shanghai, China*

ARTICLE INFO

Article history:
Received 16 August 2015
Received in revised form
17 October 2015
Accepted 6 November 2015
Communicated by Jinhui Tang
Available online 2 December 2015

Keywords:
Generic multiple task learning
Flexible structure regularization
Joint ℓ_{11}/ℓ_{21}-norm regularization
Iteratively reweighted least square
Accelerated proximal gradient

ABSTRACT

Due to the theoretical advances and empirical successes, *Multi-task Learning* (MTL) has become a popular design paradigm for training a set of tasks jointly. Through exploring the hidden relationships among multiple tasks, many MTL algorithms have been developed to enhance learning performance. In general, the complicated hidden relationships can be considered as a combination of two key structural elements: task grouping and task outlier. Based on such task relationship, here we propose a *generic* MTL framework with *flexible structure regularization*, which aims in relaxing any type of specific structure assumptions. In particular, we directly impose a joint ℓ_{11}/ℓ_{21}-norm as the regularization term to reveal the underlying task relationship in a flexible way. Such a *flexible structure regularization* term takes into account any convex combination of grouping and outlier structural characteristics among the multiple tasks. In order to derive efficient solutions for the generic MTL framework, we develop two algorithms, i.e., the Iteratively Reweighted Least Square (IRLS) method and the Accelerated Proximal Gradient (APG) method, with different emphasis and strength. In addition, the theoretical convergence and performance guarantee are analyzed for both algorithms. Finally, extensive experiments over both synthetic and real data, and the comparisons with several state-of-the-art algorithms demonstrate the superior performance of the proposed generic MTL method.

1. Introduction

Realizing the existence of sparse training data and the task correlations, multiple task learning (MTL) is designed to train multiple models jointly and simultaneously, and often leads to better learnt models than those trained independently. The key idea of MTL is to explore the hidden relationships among multiple tasks to enhance learning performance. MTL has been shown particularly useful if there exist intrinsic relationships among multiple learning tasks and the training data is inadequate for each single task. Due to its empirical successes, MTL has been applied to various application domains, including social media categorization and search [12,54], fine-grained visual categorization [42], disease modeling and prediction [8,63], spam filtering [3], reinforcement learning [9] and even financial stock selection [19].

The key ingredient of MTL is to explore *model commonality* among the multiple learning tasks, and use such model commonality to improve the learning performance. Some earlier MTL work assume that there is a common structure or a common set of parameters shared by all the learning tasks [51,52]. However, sharing a model commonality among all the learning tasks is a fairly strong assumption, which is often invalid in real applications. Therefore, two compromised yet more realistic scenarios, i.e., *task grouping* and *task outlier*, have been explored recently. For task grouping, one assumes that the commonality only exists among tasks within the same group. During the learning process, through identifying such task groups, the unrelated tasks from different groups will not influence each other [21,23,52,55]. In the task outlier scenario [13], a robust MTL algorithm was proposed to capture the commonality for a major group of tasks while detecting the outlier tasks. A popular way to tackle the robust MTL problem is to use a decomposition framework, which forms the learning objective with a structure term and an outlier penalty term. To efficiently solve the optimization problem, the target model can be further decomposed into two components, reflecting the major group structure and the outliers [22]. Representative decomposition schemes for MTL include the low-rank structure [13] and the group sparsity based approaches [20].

Note that the aforementioned assumptions of *task grouping* and *task outlier* were exclusively considered in most of the existing works. In other words, the *task grouping* based methods neglected the existence of outlier tasks and many robust MTL frameworks

[*] Corresponding author.
E-mail address: jianpu@ion.ac.cn (J. Pu).

http://dx.doi.org/10.1016/j.neucom.2015.11.029
0925-2312/© 2015 Elsevier B.V. All rights reserved.

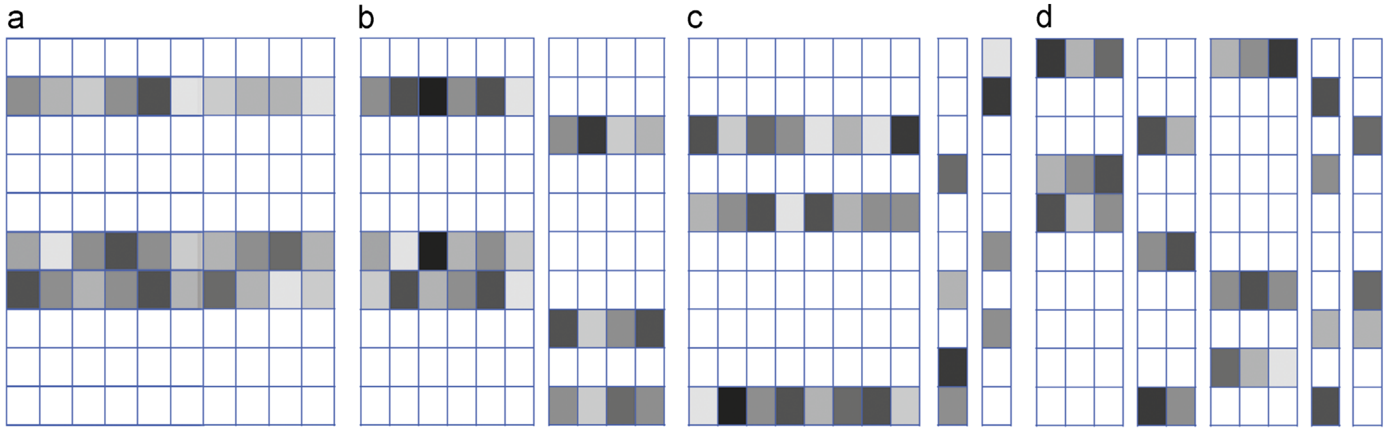

Fig. 1. The illustration of different target models **W** learned using various assumptions of task structures: (a) shared model commonality, (b) task grouping, (c) outlier tasks, and (d) generic multi-tasks. Each column of **W** is corresponding to a single task and each row represents a feature dimension. For each element in **W**, white color means zero-valued elements and gray color indicates non-zero values with the intensity indicating the magnitude of the values.

only assumed the case of one major task group peppered with a few outlier tasks. In this paper, we address MTL under a very general setting where multiple major task groups and outlier tasks could occur simultaneously. In particular, without decomposing the target model, we directly impose a *flexible structure regularization* term with a joint ℓ_{11}/ℓ_{21}-norm that reflects a mixture of structure and outlier penalties. The final objective is formulated as an unconstrained non-smooth convex problem and two efficient algorithms, i.e., the Iteratively Reweighted Least Square (IRLS) method and the Accelerated Proximal Gradient (APG) method, are applied to derive optimal solutions with different strength. Particularly, the IRLS method can handle the learning process for a large number of tasks efficiently, while the APG method provides robust performance when the active features are either sparse or dense. In addition, we provide theoretical analysis on both convergence and performance bound of the proposed MTL method. Finally, empirical studies on synthetic and real benchmark datasets corroborate that the proposed MTL learning method clearly outperforms several state-of-the-art MTL approaches.

The remainder of this paper is organized as follows. Section 2 briefly reviews several major MTL schemes in the existing works. Section 3 presents our proposed generic MTL framework and two efficient solutions. Sections 4 and 5 provide theoretical analysis of the proposed methods, including convergence properties and performance bounds. Section 6 gives experimental validations and comparative studies, and, finally, Section 7 concludes this paper.

2. Related work

Here, we first define notations used in this paper. Then we briefly survey several major multi-task learning paradigms and summarize their strengthness and weakness.

2.1. Notations

Assume the data is represented as a matrix $\mathbf{X} \in \mathbb{R}^{d \times n}$, where the column vector $\mathbf{x}_i \in \mathbb{R}^d$ is the i-th data point and d is the dimension. In addition, we denote $\mathbf{x}_{i\cdot}$ as the i-th row of $\mathbf{X}$, which corresponds to the i-th feature of the data. The norm of matrix $\mathbf{X}$ is denoted as $\|\mathbf{X}\|_{p,q} = (\sum_i \|\mathbf{x}_{i\cdot}\|_p^q)^{1/q} = (\sum_i (\sum_j x_{ij}^p)^{q/p})^{1/q}$. For example, $\|\mathbf{X}\|_{2,1} = \sum_i \|\mathbf{x}_{i\cdot}\|_2 = \sum_i (\sum_j x_{ij}^2)^{1/2}$.

In a typical setting of multiple task regression or classification, we are given L tasks associated with training data $\{(\mathbf{X}_1, \mathbf{y}_1), \ldots, (\mathbf{X}_L, \mathbf{y}_L)\}$, where $\mathbf{X}_l \in \mathbb{R}^{d \times n_l}, \mathbf{y}_l \in \mathbb{R}^{n_l}$ are the input and response of the l-th task with a total of n_l samples. We want to employ MTL to derive optimal prediction models for all the tasks simultaneously. In particular, for linear regression models, the prediction model for the l-th task is represented as $f(\mathbf{w}_l, \mathbf{X}_l) = \mathbf{X}_l^\top \mathbf{w}_l$. We then use a coefficient matrix $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_L]$ to represent all the regression tasks. The goal of MTL is to derive an optimal $\mathbf{W}^*$ across all the learning tasks, and meanwhile satisfying desired structure characteristics.

2.2. Shared model commonality

One of the straightforward ways for designing a MTL algorithm is to assume that all the tasks share certain model commonality. Typically, such commonality can be represented as shared common structures or parameters by the learned models. For instance, structure commonality includes subspace sharing [28,35,40] and feature set sharing [2,24,31,32,34,56,61,18]. In terms of the parameter commonality, it includes a wide range of options depending on the used learning methods, such as the hidden units in neural networks [10], kernels [17], the priors in hierarchical Bayesian models [4,49,57,58,60], the parameters in Gaussian process covariance [26,43,50], the feature mapping matrices [1], and the similarity metrics [39,59]. Fig. 1(a) demonstrates an example of the feature sharing among the learned model **W**, where all the learning tasks select the same subset of features. Through exploring various types of model commonalities, either structures or parameters, simultaneously learning multiple tasks will benefit from the learning of each other. Hence, the MTL paradigm is expected to achieve better generalization performance than independently learning a prediction model for each task. However, the real applications tend to have more complicated situations and often there are not commonly shared structure or parameters among all the tasks [47,38].

2.3. MTL with task grouping

Note that in many real applications for learning multi-tasks, the tasks are gathered into several groups according to their relatedness. Intuitively, the tasks in the same group are more related than the tasks in different groups. As shown in Fig. 1(b), the learning tasks form two groups, where the tasks within the same group select the same subset of features and share no common features with the tasks from the other group.

To deal with this scenario, one of the representative methods is to use grouping matrices to model the *task grouping* effect