Noname manuscript No.
(will be inserted by the editor)

A Brief Review on Multi-Task Learning

Kim-Han Thung · Chong-Yaw Wee

the date of receipt and acceptance should be inserted later

Abstract Multi-task learning (MTL), which optimizes multiple related learning tasks at the same time, has been widely used in various applications, including natural language processing, speech recognition, computer vision, multimedia data processing, biomedical imaging, socio-biological data analysis, multi-modality data analysis, etc. MTL sometimes is also referred to as joint learning, and is closely related to other machine learning subfields like multi-class learning, transfer learning, and learning with auxiliary tasks, to name a few. In this paper, we provide a brief review on this topic, discuss the motivation behind this machine learning method, compare various MTL algorithms, review MTL methods for incomplete data, and discuss its application in deep learning. We aim to provide the readers with a simple way to understand MTL without too many complicated equations, and to help the readers to apply MTL in their applications.

Keywords Multi-task learning, MTL, transfer learning, joint learning, multi-class learning, learning with auxiliary tasks

1 Introduction

1.1 What is a task?

A task is generally referred to the learning of an output target using a single input source. If the input source consists of a single variable (or feature), we will have a univariate analysis, if the input source consists of multiple variables

Kim-Han Thung
Dept. of Radiology, University of North Carolina, Chapel Hill, U.S.A.
khthung@med.unc.edu

Chong-Yaw Wee (Corresponding Author)
Dept. of Biomedical Engineering, National University of Singapore, Singapore
cywee2000@gmail.com

(or features), we will have a multivariate analysis. In this sense, "multiple tasks" could mean the learning of multiple output targets using a single input source, or the learning of single output target using multiple input sources, or a mixture of both. Depending on the definition of "multiple tasks", the multi-task learning (MTL) could have different objective functions, as we will demonstrate in the following subsections.

1.2 What is multi-task learning (MTL) and why it is useful?

When the original data representation is high dimensional and the number of examples provided to solve a regression or classification problem is limited, any learning algorithm which does not use any sort of prior knowledge will perform poorly due to lack of data to reliably estimate the model parameters. This issue is particularly crucial for applications such as medical image analysis since it needs more manual labor to label data instances. Multi-task learning (MTL) [72, 5, 19], as one type of machine learning method that aims to solve multiple tasks simultaneously, can be a good recipe by exploiting useful information from other related learning tasks to help alleviate this data scarcity problem. Rich Caruana [8] has summarized the goal of MTL succinctly: "*MTL is an approach to inductive transfer that improves generalization by using the domain information contained in the training signals of related tasks as an inductive bias. It does this by learning tasks in parallel while using a shared low dimensional representation; what is learned for each task can help other tasks be learned better*". The basic assumption of MTL is that all the tasks in learning, or at least a subset of them, are associated with each other, and thus the shared information among different tasks can lead to better learning performance if all the tasks are learned jointly, comparing to learn them independently. In other words, it assumes that the learning of one task can improve the learning of the other tasks. This is generally achieved by learning all the tasks jointly, utilizing the correlated information among different tasks (which we will describe in more details in the following sections) to improve the learning of each task. Although MTL is particularly helpful if the tasks share significant commonalities, it has also been shown to be beneficial for learning unrelated tasks [68].

1.3 Some application examples of MTL

We describe some application examples of MTL in this subsection. MTL has been used to tackle feature selection problem that involves multi source data (please refer to Section 2.1.1 for the formulation). In this case, not all attributes (or features) in multi-source data are useful for classification or regression problem. When the features derived from different data sources are related in some ways (e.g., when they are originated from the same regions of interest and thus correspond to each other), then it is better to select the features from all the data sources jointly by using a joint selection regularizer. Some of the regularizers (or constraints) that have been introduced include joint sparsity, low-rank, graph sparse coding, graph self-representation, and so on [116,32,111,106,100,112]. The recent studies show that the inclusion of these additional regularizations during joint feature selection (via MTL) can improve the performance of their classification models if compared with the model that selects features from each data source individually.

MTL has also been used to deal with neurodegenerative disease diagnosis problem. For example, we can use structural Magnetic Resonance Imaging (sMRI) data to predict the values for different kind of clinical scores and the diagnostic label of a subject. For Alzheimer's disease (AD) studies, the clinical scores that are usually used to grade the healthiness and functionality of a human brain include the Mini-Mental State Examination (MMSE), Dementia Rating Scale (DRS), Alzheimer's Disease Assessment Scale (ADAS), etc. In this case, the prediction of one of the target output using sMRI data is a learning task. As all the target outputs (e.g., clinical scores and diagnostic label) are related, learning all the tasks together (using MTL strategy) would most probably give us a better prediction results than the results obtained from learning all the tasks independently (please refer to Section 2.1.2 for the formulation) [74,85,114].

Furthermore, MTL has also been used in self-driving automation system, as described in [59]. Using the image acquired from the camera attached to the car, we would like to detect different objects on the road, e.g., pedestrian, car, road sign, traffic light, etc. We can use neural network architecture that takes camera image as input, and the object labels (e.g., pedestrian, car, etc.) as outputs. When we train this neural network to learn multiple objects simultaneously, it becomes a MTL problem. In this case, we hope that building a single neural network to learn multiple objects is better than learning separate neural network for each object (please refer to Section 4 for more details). In fact, learning a set of tasks together in neural network has many advantages, e.g., we could have more training samples, we can learn shared lower-level features (e.g., edge, line, and shape), and the learning of one task could benefit from the learning from the other tasks.

1.4 Related research subfields

MTL is very similar to inductive transfer learning [63,86], where there are two types of tasks, i.e., the primary task, which is the main goal of the study, and the secondary (or auxiliary) task, which is associated with or related to the main task, but is not the main goal of the study. In transfer learning, it is assumed that the associated secondary task can provide extra information to the primary task, and improves the generalization of the main task. However, there is no such distinction for MTL, as it generally treats all the tasks equally. Therefore, transfer learning can be thought of as a special case of MTL.

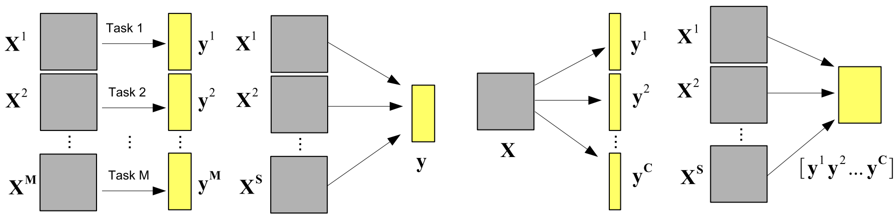
Besides, MTL is also related to the problem of learning-to-learn (LTL), which aims to perform a new task by exploiting the knowledge acquired when solving previous tasks. The capability of LTL is very similar to the ability of human being that learns from experience when performing new tasks. Hence, a solution or machine constructed based on LTL would have major impact in general Artificial Intelligence (AI). Recently, learning nonlinear hierarchical representations from multiple tasks using multilayer deep neural networks [71] has emerged as one of the hottest research topics in AI. Researchers have shown improved results in a number of empirical domains, particularly in computer vision [24]. This success has increased interest in multi-task representation learning in deep neural networks, as we will discuss in more details in Section 4. In the following sections, we begin our discussion by introducing different formulations of MTL algorithms.

2 Formulation of MTL Algorithms

The typical formulation for a conventional MTL algorithm [8, 5, 72, 102] is given in the following form:

$$\min_{\mathbf{W}=[\mathbf{w}^1\mathbf{w}^2\cdots\mathbf{w}^M]} \sum_{m=1}^{M} L(\mathbf{X}^m, \mathbf{y}^m, \mathbf{w}^m) + \lambda \mathrm{Reg}(\mathbf{W}), \tag{1}$$

where $\mathbf{X}^m \in \mathbb{R}^{N_m \times D}$ denotes the input matrix of the m-th task, $\mathbf{y}^m \in \mathbb{R}^{N_m \times 1}$ denotes the corresponding m-th task output vector, and $\mathbf{w}^m \in \mathbb{R}^{D \times 1}$ denotes the weight vector (or regression parameters) for the m-th task that maps $\mathbf{X}^m$ to $\mathbf{y}^m$, e.g., $\mathbf{y}^m \approx \mathbf{X}^m\mathbf{w}^m$ (for regression problem). The scalars N_m, D, and M denote the number of samples for the m-th task, the number of features for each input matrix, and the number of tasks, respectively. Note that at this stage, we assume that all the input matrices in $\{\mathbf{X}^m, m = 1, 2, \cdots, M\}$ are having the same dimensionality of features (but can have different number of samples for each task), and the features of all the tasks are corresponding, so that we can concatenate all the weight vectors in $\{\mathbf{w}^m\}$ together to obtain $\mathbf{W} = [\mathbf{w}^1\mathbf{w}^2 \cdots \mathbf{w}^M]$ (i.e., features in each row of $\mathbf{W}$ are corresponding). Based on the prior knowledge of the data and different assumptions on the relationship among tasks, we can design different constraints for $\mathbf{W}$, which is normally implemented in the regularizer of $\mathbf{W}$, denoted by $\mathrm{Reg}(\mathbf{W})$. In addition, λ is the regularization parameter that controls the balance between the loss function (first term) and the regularizer (second term) in Eq. (1). If we set the value of λ to zero, we will have a solution of $\mathbf{W}$ that does not use any assumption or prior knowledge about the task relatedness, which will most probably only perform well on the training data, but not on the testing data (i.e., data overfitting). In contrary, if we set the value of λ too high, we may have a general solution of $\mathbf{W}$ that satisfies the task-relatedness assumption given in $\mathrm{Reg}(\mathbf{W})$, but does not perform well for all the prediction tasks. Thus, the regularization parameter (and any other hyper-parameters) is normally determined via inner cross-validation using the training samples.



(a) MTL **(b)** MISO **(c)** SIMO **(d)** MIMO

Fig. 1: (a) General form of Multi-task learning (MTL), and its special cases, i.e., (b) multi-input single-output (MISO), where multiple sets of input are mapped to a same set of output target, (c) single-input multi-output (SIMO), where one set of input map is mapped to multiple sets of output target, and (d) multi-input multi-output (MIMO), where multiple sets of input are mapped to the same multiple sets of output target. ($\mathbf{X}$: Input data; $\mathbf{Y}$: Target output)

In summary, Eq. (1) consists of two terms, i.e., 1) the data fidelity term (the first term in Eq. (1)), which computes how well the target predictions match with the ground truth targets, and 2) the regularization term (i.e., the second term in Eq. (1)), which regularizes the weight matrix $\mathbf{W}$ to garner the relationship among different learning tasks, to have better prediction model for each learning task [3, 6, 8, 18, 85, 117, 118]. In the following subsections, we describe these two terms in more details.

2.1 Different data fidelity terms of MTL

Let us examine the data fitting term in Eq. (1), it seems that there are M number of input matrices and output vectors for M number of tasks. However, in many real life applications, some of the input matrices or output vectors are shared among different tasks. Based on the number of unique input in $\{\mathbf{X}^m\}$ and unique output in $\{\mathbf{y}^m\}$, we categorize the MTL problems into three special cases, namely the multi-input single-output (MISO), the single-input multi-output (SIMO), and the multi-input multi-output (MIMO). Fig. 1 shows the general form of MTL and its three special cases based on whether the inputs or outputs are shared among different tasks. Each special case of MTL would have slightly different form of data fitting term in Eq. (1). In addition, the definition of task, and how it relates the input data to the output target would also be slightly different. We will discuss each of these special cases in more details in the following subsections.

2.1.1 Multi-Input Single-Output (MISO)

In this case, we have, for example, multiple data sources, and all are mapped to the same target $\mathbf{y}$. A task is thus defined as the prediction of one data source to a common target $\mathbf{y}$. For instance, this can happen if we have multi-modality data, and each modality can be used to predict the target $\mathbf{y}$. Another example in multimedia application is that there are multiple views of the same object, and all the views are used jointly to predict the label of this object. The mean square loss formulation for the data fidelity term in this case is given as:

$$L(\mathbf{X}, \mathbf{y}, \mathbf{W}) = \sum_{s}^{S} \|\mathbf{X}^s \mathbf{w}^s - \mathbf{y}\|_F^2 \ , \tag{2}$$

where $\mathbf{X} = \{\mathbf{X}^1, \mathbf{X}^2, \cdots, \mathbf{X}^S\}$ denotes the set of multi-source (also known as multi-view or multi-modality) data, $\mathbf{X}^s \in \mathbb{R}^{N \times D}$ denotes the s-th data source, $\mathbf{y} \in \mathbb{R}^{N \times 1}$ denotes the output feature vector, and $\mathbf{W} = [\mathbf{w}^1 \mathbf{w}^2 \cdots \mathbf{w}^S] \in \mathbb{R}^{D \times S}$ denotes the weight matrix with its s-th column $\mathbf{w}^s$ denoting the weight vector corresponding to the mapping of $\mathbf{X}^s$ to $\mathbf{y}$. The N, S and D denote the number of samples, the number of data sources, and the number of features in each data source, respectively. Note that $\hat{\mathbf{y}}^s = \mathbf{X}^s \mathbf{w}^s$ is the prediction of $\mathbf{y}$ for the s-th task. Without loss of generality, the bias terms are omitted in this review paper, as they can easily be incorporated into Eq. (2) by adding a column of ones to $\mathbf{X}^s$. Examples of research works using this setting include [47,48].

Other than mean square loss function, which is suitable for regression task, the data fidelity term can also take the form of logistic or hinge loss function for classification task, which are respectively given below as

$$L(\mathbf{X}, \mathbf{y}, \mathbf{W}) = \sum_{s}^{S} \sum_{j}^{N} \log(1 + \exp(-\hat{y}_j^s y_j)), \tag{3}$$

$$L(\mathbf{X}, \mathbf{y}, \mathbf{W}) = \sum_{s}^{S} \sum_{j}^{N} \max(0, 1 - \hat{y}_j^s y_j), \tag{4}$$

where $\hat{y}_j^s = \mathbf{x}_j^s \mathbf{w}^s$ denotes the prediction of the j-th sample of the s-th data source ($\mathbf{x}_j^s$), and $y_j \in \{-1, 1\}$ is the corresponding ground truth label.

2.1.2 Single-Input Multi-Output (SIMO)

In this case, there is only one input (or all the tasks share the same input), and it is used to predict different types of output target. A task is thus defined as the prediction of the input matrix $\mathbf{X}$ to a target vector. For example, given an image, we would like to predict what are the contents of this image. The mean square loss function for the data fidelity term in this case is given as:

$$L(\mathbf{X}, \mathbf{Y}, \mathbf{W}) = \sum_{c}^{C} \|\mathbf{X}\mathbf{w}^c - \mathbf{y}^c\|_F^2 = \|\mathbf{X}\mathbf{W} - \mathbf{Y}\|_F^2 \ , \tag{5}$$

where $\mathbf{X} \in \mathbb{R}^{N \times D}$ and $\mathbf{Y} = [\mathbf{y}^1 \cdots \mathbf{y}^C] \in \mathbb{R}^{N \times C}$ are the input feature matrix and output target matrix, respectively, while $\mathbf{W} = [\mathbf{w}^1 \mathbf{w}^2 \cdots \mathbf{w}^C] \in \mathbb{R}^{D \times C}$ are the corresponding weight matrix, with its c-th column is the weight vector $\mathbf{w}^c$

that corresponds to the prediction of $\mathbf{y}^c$ using $\mathbf{X}$. As all the learning tasks share the same input samples, this multi-task learning for classification problem is also called multi-class learning [47,48,46,36,60,76]. The logistic and hinge loss functions for this setting are similar to Eq. (3) and Eq. (4), respectively, by replacing the sum over the number of data sources with the sum over the number of targets.

2.1.3 Multi-Input Multi-Output (MIMO)

In MIMO, multiple input sources are used to predict multiple output targets. A task here is defined as the prediction of one input source to a single target. For example, this can happen if we have multiple modalities of data, and each modality can be used to predict several target labels (i.e., multi class classification). The mean square loss function for the data fidelity term of a MIMO problem is given as:

$$L(\mathbf{X},\mathbf{Y},\mathbf{W}) = \sum_{s}^{S}\sum_{c}^{C} \|\mathbf{X}^s\mathbf{w}^{s,c} - \mathbf{y}^c\|_F^2 = \sum_{s}^{S} \|\mathbf{X}^s\mathbf{W}^s - \mathbf{Y}\|_F^2 \ , \qquad (6)$$

where $\mathbf{X} = \{\mathbf{X}^1, \mathbf{X}^2, \cdots, \mathbf{X}^S\} \in \mathbb{R}^{N\times D}$, as defined in Section 2.1.1, $\mathbf{Y} = [\mathbf{y}^1\mathbf{y}^2\cdots\mathbf{y}^C] \in \mathbb{R}^{N\times C}$, as defined in Section 2.1.2, $\mathbf{W} = \{\mathbf{W}^1, \mathbf{W}^2, \cdots, \mathbf{W}^S\} \in \mathbb{R}^{D\times C}$ is the set of all the weight matrices, and $\mathbf{W}^s = [\mathbf{w}^{s,1}\mathbf{w}^{s,2}\cdots\mathbf{w}^{s,C}] \in \mathbb{R}^{D\times C}$ is the weight matrix for the s-th modality data $\mathbf{X}^s$. The derivations of logistic and hinge loss functions for this setting are straight forward and similar to Eq. (3) - (4). In some applications, the multi-source data are concatenated into a single input data, which simplify this problem to SIMO (Section 2.1.2). Examples of research works with this setting include [94,47,48].

2.2 Different regularizations on weight matrix $\mathbf{W}$ of MTL

In this section, we adopt the general notation of loss function as in Eq. (1), and focus on discussing some of the commonly used constraints on the weight matrix $\mathbf{W}$ of the MTL formulation in Eq. (1).

2.2.1 MTL with Lasso

Assuming there is little correlation among different tasks, and we know that the weight matrix should be sparse for better interpretability and accuracy of the results, the multi-task learning with Lasso constraint [75] is given as below via ℓ_1-norm regularization on $\mathbf{W}$:

$$\min_{\mathbf{W}} \sum_{m=1}^{M} L(\mathbf{X}^m, \mathbf{y}^m, \mathbf{w}^m) + \lambda\|\mathbf{W}\|_1, \qquad (7)$$

where $\mathbf{W} = [\mathbf{w}^1\mathbf{w}^2 \cdots \mathbf{w}^M]$, and λ is the regularization parameter that controls sparsity in $\mathbf{W}$. Higher value of λ would correspond to a sparser model, i.e., more zero-value elements in $\mathbf{W}$. Lasso constraint makes sense when not all the features in the tasks are useful, and there are only weak associations among tasks. As Lasso constraint does not fully utilize the task relatedness, it is also usually used in combination with other constraints. Works that based on the variations of this constraint include [51,45,110,52,43].

2.2.2 MTL with group sparsity constraint

$\boldsymbol{\ell_{2,1}}$**-norm**. One direct way to extract the task relatedness information from multiple related tasks is to constrain all models to share a common set of features. This goal is accomplished by solving the following $\ell_{2,1}$-norm regularized MTL problem [5,6]

$$\min_{\mathbf{W}} \sum_{m=1}^{M} L(\mathbf{X}^m, \mathbf{y}^m, \mathbf{w}^m) + \lambda \|\mathbf{W}\|_{2,1}, \tag{8}$$

where $\|\mathbf{W}\|_{2,1} = \sum_{i=1}^{D} \|\mathbf{w}_i\|_2$, with $\mathbf{w}_i \in \mathbb{R}^{1\times M}$ denotes i-th row in $\mathbf{W}$. Note that the $\ell_{2,1}$-norm ($\|\cdot\|_{2,1}$) enforces row-wise sparsity, i.e., it encourages all-zero-value rows in $\mathbf{W}$ for $\|\mathbf{W}\|_{2,1}$. This is equivalent to joint feature selection for all the learning tasks. Research works that use Eq. (8) include [60,47,61,28, 84,51], where different optimization algorithms have been proposed to solve it. Some variates for this MTL formulation include [40], where weighted $\ell_{2,1}$-norm is proposed, and [67], where feature groups can overlap with each other.

$\boldsymbol{\ell_{p,q}}$**-norm**. Eq. (8) can be generalized to use $\ell_{p,q}$-norm regularizer to select features, as given by

$$\min_{\mathbf{W}} \sum_{m=1}^{M} L(\mathbf{X}^m, \mathbf{y}^m, \mathbf{w}^m) + \lambda \|\mathbf{W}\|_{p,q}, \tag{9}$$

where $\|\mathbf{W}\|_{p,q} = \|[\|\mathbf{w}_1\|_p \cdots \|\mathbf{w}_i\|_p \cdots \|\mathbf{w}_D\|_p]\|_q$. Eq. (9) is convex if $p > 1, q \geq 1$. MTL algorithms that use Eq. (9) include [45,77,58] that uses $\ell_{\infty,1}$-norm, and [79] that uses $\ell_{p,1}$-norm.

Capped $\boldsymbol{\ell_{p,1}}$**-norm**. In order to obtain a sparser subset of features, Gong *et al.* [26] proposed a capped $\ell_{p,1}$-norm penalty for $\mathbf{w}_i$, as given by

$$\min_{\mathbf{W}} \sum_{m=1}^{M} L(\mathbf{X}^m, \mathbf{y}^m, \mathbf{w}^m) + \lambda \sum_{i=1}^{D} \min(\|\mathbf{w}_i\|_p, \theta), \tag{10}$$

where θ is a threshold value. The capped $\ell_{p,1}$-norm in Eq. (10) causes the algorithm to focus on minimizing the rows of $\mathbf{W}$ with ℓ_p-norm less than θ, thus encouraging sparser solution. In other words, the smaller the value of θ, the sparser is the solution of Eq. (10), and vice versa. When the value of θ is large enough, the capped $\ell_{p,1}$-norm in Eq. (10) will become $\ell_{p,1}$-norm.

Multi-level Lasso. Up to this point, the regularization is imposed on the weight matrix $\mathbf{W}$ directly to garner the task relatedness. Another line of research decomposes the weight matrix into several components and imposes different regularization on them. By using the right regularizers on these components, we can get similar group sparsity effect induced by $\ell_{p,q}$-norm of weight matrix $\mathbf{W}$. For example, Lozano *et al.* [52] multi-level Lasso, given as

$$\min_{\tilde{\mathbf{W}},\boldsymbol{\theta}} \sum_{m=1}^{M} L(\mathbf{X}^m, \mathbf{y}^m, \mathbf{w}^m) + \lambda_1 \|\boldsymbol{\theta}\|_1 + \lambda_2 \|\tilde{\mathbf{W}}\|_1, s.t. \mathbf{W} = \mathrm{diag}(\boldsymbol{\theta})\tilde{\mathbf{W}}, \qquad (11)$$

where $\boldsymbol{\theta} \in \mathbb{R}^{D\times 1}$ is a non-negative coefficient vector that controls the feature-level group sparsity. When one of the elements in $\boldsymbol{\theta}$ (*e.g.,* θ_i) is zero, then the corresponding feature weight (*e.g.,* $\mathbf{w}_i$, i-th row in $\mathbf{W}$) would be zero as well, resulting group sparsity effect on $\mathbf{W}$. It was shown in [52] that the regularizer in Eq. (11) is equivalent to $\ell_{1,\frac{1}{2}}$. Wang *et al.* generalized Eq. (11) by using $\ell_{p,1}$-norm for $\tilde{\mathbf{W}}^\top$ (i.e., transpose of $\tilde{\mathbf{W}}$) and ℓ_q-norm for $\boldsymbol{\theta}$. Other variations of Eq. (11) include [31,36,37,38,49].

Structured group Lasso. On the other hand, instead of assuming all tasks are related using $\ell_{p,1}$-norm on the weight matrix $\mathbf{W}$, some studies, such as [38,49], utilize the prior knowledge of the data and task relatedness to impose group sparsity on the same group of tasks. More specifically, the regularizer of $\mathbf{W}$ is given by $\sum_{i=1}^{D} \sum_{v\in V} \lambda_v \|\mathbf{w}_i^{G_v}\|_2$, where V denotes the number of group, and G_v denotes a set of related tasks in group v. In this way, only tasks within the same group will have joint sparsity constraint and may share a common set of relevant input features, while the weakly related tasks (tasks from different groups) will less likely to be affected by the same set of features. If the grouping of tasks follows hierarchical tree structure, we will have tree-structured MTL [38,29]. More specifically, in tree-guided group Lasso [38], V denotes the number of nodes in a tree (assuming a task is denoted by a node in a tree), and G_v denotes the set of tasks in the subtree rooted at node v.

Temporal group Lasso. In cases where the learning tasks involve time such as longitudinal study, there could exist temporal relationship among tasks. For example, when we predict the same taget using data at different time points, or when we use the same data to predict targets at different time points. In this case, it may be beneficial to add temporal smoothness constraint on the $\mathbf{W}$ to ensure the weight vectors at adjacent time points are consistent [109]. The MTL with temporal smoothness term is given as

$$\min_{\mathbf{W}} \sum_{m=1}^{M} L(\mathbf{X}^m, \mathbf{y}^m, \mathbf{w}^m) + \lambda_1 \|\mathbf{W}\|_F^2 + \lambda_2 \sum_{m=1}^{M-1} \|\mathbf{w}^m - \mathbf{w}^{m+1}\|_2^2, \qquad (12)$$

where the weight vectors of adjacent tasks are assumed to be similar and not to differ too much. The temporal smoothness term can be rewritten as $\|\mathbf{W}\mathbf{H}\|_F^2$, where $\mathbf{H} \in \mathbb{R}^{M\times(M-1)}$ is a matrix with $H_{ij} = 1$ if $i = j, H_{ij} = -1$ if $i = j+1,$and $H_{ij} = 0$ otherwise. The variants of MTL that utilize temporal smoothness prior include sparse fused group Lasso [108], variant of fused Lasso [105], etc [89,83,41,91].

2.2.3 MTL with low-rank constraint

Instead of using group sparsity constraint, another way to extract task relationship is to constrain the prediction models from different tasks to share a low-dimensional subspace, i.e., $\mathbf{W}$ is of low rank. This type of MTL can be accomplished by solving the following surrogate approximation of rank minimization problem [22,64]:

$$\min_{\mathbf{W}} \sum_{m=1}^{M} L(\mathbf{X}^m, \mathbf{y}^m, \mathbf{w}^m) + \lambda \|\mathbf{W}\|_*, \tag{13}$$

where $\|\cdot\|_*$ denotes the trace norm (or nuclear norm), i.e., the sum of the singular values, $\|\mathbf{W}\|_* = \sum_{i=1}^{\min(M,D)} \sigma_i(\mathbf{W})$. There are some variations of trace norm regularization such as [26,30], which uses a capped trace regularizer $\sum_{i=1}^{\min(M,D)} \min(\sigma_i(\mathbf{W}), \theta)$ that penalizes only small singular values of $\mathbf{W}$ that determine the rank of $\mathbf{W}$, and [56] that introduces spectral k-support norm. There are also other ways to formulate low rank constraint rather than nuclear norm, for example by finding two low-rank matrices, and let $\mathbf{W}$ equal to the product of these two matrices. We will discuss this case in more details in Section 2.3.2.

2.2.4 MTL for unrelated tasks

Some studies have shown that unrelated groups of tasks can be exploited to improve the learning of certain task [17,78,82]. Thus, it is still possible that the MTL can be beneficial to the main task even if the other tasks (e.g., auxiliary tasks) are not related to the main task. For example, if we have prior knowledge that the tasks are not related, we can have regularization that promotes task exclusiveness. In [110], Zhou et al. assume that all tasks are exclusive and use $\ell_{1,2}$-norm to regularize $\mathbf{W}$, i.e., $\|\mathbf{W}\|_{1,2}^2$. In [68], Bernardino *et al.* introduce a regularization term that penalizes the inner product between the weight matrices of two different groups of tasks (details in Sec. 2.3.2).

2.2.5 MTL with graph Laplacian regularization

Other than regularization that utilizes the relationship among tasks, which normally assumes that the weight parameters for related task are similar, we can also use graph level regularization that utilizes the relationship among samples [115]. Specifically, we can introduce a graph Laplacian regularization to preserve the local topological relation among samples, i.e., the local structural information of the sample is preserved after the transformation using the weight matrix $\mathbf{W}$. The graph Laplacian regularization is given as

$$\sum_{i,j}^{N} s_{ij}\|\mathbf{x}_i\mathbf{W} - \mathbf{x}_j\mathbf{W}\|_2^2 = tr(\mathbf{W}^\top\mathbf{X}^\top\mathbf{L}\mathbf{X}\mathbf{W}), \tag{14}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{S} \in \mathbb{R}^{N\times N}$ denotes the Laplacian matrix, $\mathbf{S} = [s_{ij}] \in \mathbb{R}^{N\times N}$ denotes the similarity matrix between every pair of sample points $\mathbf{x}_i$ and $\mathbf{x}_j$, and $\mathbf{D} \in \mathbb{R}^{N\times N}$ is a diagonal matrix with its diagonal element defined as $d_{ii} = \sum_j s_{ij}$.

2.3 Different regularizers for decomposed weight matrix $\mathbf{W}$ of MTL

Recently, more advanced MTL algorithms have been proposed by decomposing the weight matrix $\mathbf{W}$ in Eq. (1) into summation or/and product of two or more matrices. For example, in MTL dirty model [35], the weight matrix is decomposed into the summation of two matrices (i.e., $\mathbf{W} = \mathbf{P} + \mathbf{Q}$), each with different constraint to model a more complicated relationship among tasks. In [6], the weight matrix is decomposed into a product of two matrices (i.e., $\mathbf{W} = \mathbf{B}\mathbf{A}$), also each with different constraint, to enable feature transformation while performing MTL. In general, the weight matrix $\mathbf{W}$ can be decomposed into

$$\mathbf{W} = \mathbf{P} + \mathbf{B}\mathbf{A}, \tag{15}$$

where $\mathbf{P}$ and $\mathbf{A}$ can be seen as the coefficient matrix in the original and the transformed feature space, respectively, while $\mathbf{B}$ is the transformation matrix. In the following, we discuss these cases in more details.

2.3.1 MTL with $\mathbf{W} = \mathbf{P} + \mathbf{Q}$

Dirty model. MTL based on group sparsity penalization (ℓ_1/ℓ_q-norm regularization) performs well in ideal cases or applications in which the prediction parameters of all the learning tasks share the same structure, i.e., they share the same features. However, simply using the ℓ_1/ℓ_q-norm regularization may not be effective for many practical applications that deal with prediction tasks that may not fall into a single structure. This can be accomplished by decomposing $\mathbf{W}$ into two components $\mathbf{P}$ and $\mathbf{Q}$ while solving the dirty multitask least squares problem [35]

$$\min_{\mathbf{P},\mathbf{Q}} \sum_{m=1}^{M} L(\mathbf{X}^m, \mathbf{y}^m, (\mathbf{p}^m + \mathbf{q}^m)) + \lambda_1\|\mathbf{P}\|_{1,\infty} + \lambda_2\|\mathbf{Q}\|_1, \tag{16}$$

where $\mathbf{P} = [\mathbf{p}^1 \cdots \mathbf{p}^M] \in \mathbb{R}^{D\times M}$ is the group sparsity component and $\mathbf{Q} = [\mathbf{q}^1 \cdots \mathbf{q}^M] \in \mathbb{R}^{D\times M}$ is the element-wise sparse component, λ_1 controls the group sparsity regularization on $\mathbf{P}$ and λ_2 controls the sparsity regularization on $\mathbf{Q}$. In other words, Eq. (17) encourages all the tasks to select the same set of features (via group sparsity component) while each individual task can still select features which are not common to other tasks, but are discriminative to its own task. Studies that follow this line of work include [73, 34, 107].

Robust multi-task feature learning. In [27], Gong *et al.* propose a robust multi-task feature learning, where not only row-sparsity is imposed on $\mathbf{W}$ to select common features across tasks, but column sparsity is also imposed on $\mathbf{W}$ to identify outlier tasks that do not share the same structure as the other tasks. This is acheived by letting $\mathbf{W} = \mathbf{P} + \mathbf{Q}$, and imposing $\ell_{2,1}$-norm on $\mathbf{P}$ and $\mathbf{Q}^\top$, respectively, as given below

$$\min_{\mathbf{P},\mathbf{Q}} \sum_{m=1}^{M} L(\mathbf{X}^m, \mathbf{y}^m, (\mathbf{p}^m + \mathbf{q}^m)) + \lambda_1 \|\mathbf{P}\|_{2,1} + \lambda_2 \|\mathbf{Q}^\top\|_{2,1}. \tag{17}$$

A mixture of sparse and low-rank model. Assumption in Section 2.2.3 that all models share the common low-dimensional subspace is too restrictive in some applications. To enable simultaneous learning of incoherent sparse and low-rank patterns, an extension to Eq. (13) has been proposed by decomposing the task model $\mathbf{W}$ into two components, i.e., a sparse component $\mathbf{P}$ and a low-rank component $\mathbf{Q}$. This incoherent sparse and low-rank least squares problem is given as follows [10].

$$\min_{\mathbf{W},\mathbf{P},\mathbf{Q}} \sum_{m=1}^{M} L(\mathbf{X}^m, \mathbf{y}^m, \mathbf{w}^m) + \lambda_1 \|\mathbf{P}\|_1, \quad s.t. \;\; \mathbf{W} = \mathbf{P} + \mathbf{Q}, \;\; \|\mathbf{Q}\|_* \leq \lambda_2, \tag{18}$$

where the regularization parameter λ_1 controls the sparsity of the sparse component $\mathbf{P}$, while the λ_2 regularization parameter controls the rank of $\mathbf{Q}$. In [12], ℓ_{21}-norm (rather than ℓ_1-norm) is imposed on $\mathbf{P}$, assuming the all the tasks share the same discriminative features.

2.3.2 MTL with $\mathbf{W} = \mathbf{BA}$

Multi-task feature transformation. In our previous discussions, we mainly assume that tasks are related in the original feature space. However, there are cases where this assumption does not hold, and it might be better to first transform the original features into another feature space so that the associations among different tasks can be enhanced, and the shared representations among different tasks can be learned. One formulation example of MTL with feature transformation, where $\mathbf{W} = \mathbf{BA}$, is given as [5, 6]

$$\min_{\mathbf{A},\mathbf{B}} \sum_{m=1}^{M} L(\mathbf{X}^m, \mathbf{y}^m, \mathbf{B}\mathbf{a}^m) + \lambda \|\mathbf{A}\|_{2,1}, \quad s.t. \;\; \mathbf{B}^\top \mathbf{B} = \mathbf{I}, \tag{19}$$

where $\mathbf{B} \in \mathbb{R}^{D \times D}$ is the orthogonal transformation matrix, $\mathbf{a}^m \in \mathbb{R}^{D \times 1}$ is the model parameter for the m-th task after feature transformation, and $\mathbf{A} = [\mathbf{a}^1 \cdots \mathbf{a}^M] \in \mathbb{R}^{D \times M}$ is the predictor matrix for all the tasks with row-wise sparsity constraint (via $\ell_{2,1}$-norm). Each column of $\mathbf{B}$ is one of the orthogonal transformation vectors that maps the input data (i.e., $\{\mathbf{X}^m\}$ into one of the axes in the shared representation feature space, and the orthogonality of $\mathbf{B}$ is used to prevent redundant feature representation. On the other hand, the m-th column of $\mathbf{A}$ (i.e., $\mathbf{a}^m$) is the predictor function for the m-th learning task, and the row-wise sparsity constraint on $\mathbf{A}$ is used to exploit the task relatedness by encouraging the selection of common features in all the learning tasks. Comparing Eq. (19) with Eq. (8), it can be seen that both equations are very similar, and the main difference between them is that there is an additional step of feature transformation in Eq. (19), i.e., the input $\mathbf{X}^m$ is first transformed to $\mathbf{X}^m\mathbf{B}$ before being mapped to $\mathbf{y}^m$ via $\mathbf{a}^m$.

It is interesting that Eq. (19) is equivalent to the following optimization problem [6]:

$$\min_{\mathbf{W},\mathbf{D}} \sum_{m=1}^{M} L(\mathbf{X}^m, \mathbf{y}^m, \mathbf{w}^m) + \lambda tr(\mathbf{W}^\top \mathbf{D}^{+} \mathbf{W}), \quad s.t.\ \mathbf{D} \succeq \mathbf{0}, tr(\mathbf{D}) \leq 1, \tag{20}$$

where $tr(\cdot)$ denotes the trace of a square matrix, $\mathbf{D}^{+}$ denotes the psudoinverse of $\mathbf{D}$, $\mathbf{0}$ denotes a zero vector or matrix, and $\mathbf{D} \succeq \mathbf{0}$ means that $\mathbf{D}$ is positive semidefinite. The optimal solutions of Eq. (19) and Eq. (20) are related with the following relationship: $\mathbf{W} = \mathbf{B}\mathbf{A}$, $\mathbf{D} = \mathbf{B}\ \mathrm{diag}(\|\mathbf{a}_i\|_2/\|\mathbf{A}\|_{2,1})_{i=1}^{D}\mathbf{B}^\top$. $\mathbf{D}$ can be seen as the feature covariance matrix for all the learning tasks, and when $\mathbf{W}$ is given, the analytic solution for $\mathbf{D}$ is given as $(\mathbf{W}^\top\mathbf{W})^{\frac{1}{2}}/tr((\mathbf{W}^\top\mathbf{W})^{\frac{1}{2}})$. Note that Eq. (20) is convex and easier to solve than Eq. (19), thus many studies that that based on formulation in Eq. (19) will transform their formulations into something similar to Eq. (20), so that their problems can be solved more efficiently. Furthermore, the second term in Eq. (20) can be generalized to $\lambda tr(\mathbf{W}^\top f(W)\mathbf{W})$.

Multi-task sparse coding. If $\mathbf{B}$ is an overcomplete dictionary, i.e., $\mathbf{B} = [\mathbf{b}^1 \cdots \mathbf{b}^O] \in \mathbb{R}^{D\times O}$, where $O > D$ is the number of atoms in the dictionary, we will have multi-task sparse coding [55], given as

$$\min_{\mathbf{A},\mathbf{B}} \sum_{m=1}^{M} L(\mathbf{X}^m, \mathbf{y}^m, \mathbf{B}\mathbf{a}^m) \quad s.t.\ \ \{\|\mathbf{a}^m\|_1 \leq \lambda\}_{m=1}^{M}, \{\|\mathbf{b}^j\|_2 \leq 1\}_{j=1}^{O}, \tag{21}$$

where $\mathbf{a}^m \in \mathbb{R}^{O\times 1}$ is the sparse coding for the m-th task. Note that dictionary $\mathbf{B}$ is shared by all tasks, with each of its column bounded by ℓ_2-norm, and the group sparsity constraint in Eq. (19) is different from the lasso constraint in Eq. (21). Note also that this sparse coding method is different from the conventional sparse coding strategies in the literature [62,39,53] where each input $\mathbf{X}^m$ is a sparse combination of the atoms in the dictionary, with each atom corresponds to a label, and thus the predicted label for the input $\mathbf{X}^m$ is computed as the same sparse combination of the atom labels. The multi-task learning for this type of sparse coding is beyond the scope of this review and has been explored in [44].

Multi-task with low-rank structure. Instead of using nuclear norm, we can also use the product of two low-rank matrices [113,80] to impose low-rank regularization on $\mathbf{W}$. More specifically, we set $\mathbf{W} = \mathbf{BA}$ [113,80], where $\mathbf{B}^\top \in \mathbb{R}^{r\times D}$, $\mathbf{A} \in \mathbb{R}^{r\times M}$, and $r < \min(D, M)$ is the hyper-parameter to be tuned as the rank of matrix $\mathbf{W}$. Then, we can impose $\ell_{2,1}$-norm on $\mathbf{B}$ or/and $\mathbf{A}$ to simultaneously select features and tasks during the optimization. For example, in [113],

$$\min_{\mathbf{B},\mathbf{A}} \sum_{m=1}^{M} L(\mathbf{X}^m, \mathbf{y}^m, \mathbf{B}\mathbf{a}^m) + \lambda_1 \|\mathbf{B}^\top\|_{2,1} + \lambda_2 \|\mathbf{A}\|_{2,1},\ s.t.\ r < \min(D, M). \tag{22}$$

Shared representation with unrelated task. In [68], a MTL method that favors shared low dimensional representations within each group of tasks, while imposing penalty that encourage orthogonality between representations from two groups of unrelated tasks. The formulation is given as

$$\begin{aligned} \min_{\mathbf{B},\mathbf{A},\hat{\mathbf{A}}} & \textstyle\sum_{m=1}^{M_1} L(\mathbf{X}^m, \mathbf{y}^m, \mathbf{B}\mathbf{a}^m) + \sum_{m=1}^{M_2} L(\mathbf{X}^m, \mathbf{y}^m, \mathbf{B}\hat{\mathbf{a}}^m) + \\ & \lambda_1 \|[\mathbf{A}\hat{\mathbf{A}}]\|_{2,1} + \lambda_2 \|\mathbf{A}^\top \hat{\mathbf{A}}\|_F^2,\ \ s.t.\ \mathbf{B}^\top\mathbf{B} = \mathbf{I}, \end{aligned} \tag{23}$$

where the first two terms are the data fitting loss functions for two group of tasks, the third term is used to encourage shared features between two group of tasks, while the fourth term encourages orthogonality between two group of tasks, assuming they are unrelated. Eq. (23) can be written into the equivalent form similar to Eq. (20) so that it can be solved more easily [68].

2.3.3 MTL with $\mathbf{W} = \mathbf{P} + \mathbf{BA}$

A general MTL formulation when $\mathbf{W} = \mathbf{P} + \mathbf{BA}$ is given as

$$\min_{\mathbf{P},\mathbf{B},\mathbf{A}} \sum_{m=1}^{M} L(\mathbf{X}^m, \mathbf{y}^m, (\mathbf{p}^m + \mathbf{B}\mathbf{a}^m)) + \mathrm{Reg}(\mathbf{P}, \mathbf{A}),\ s.t.\ \mathbf{B}^\top\mathbf{B} = \mathbf{I}. \tag{24}$$

In brief, this is a combination of regularizer in Section 2.3.1 and Section 2.3.2. In [3], Ando et al. assume that part of the model parameters of different tasks share a low-rank subspace, and their second term in Eq. (24) only regularizes $\mathbf{P}$. More specifically, they define $\|\mathbf{P}\|_F^2$ for the regularizer and use $\mathbf{B}$ as the shared low-rank subspace by multiple tasks. The orthonormal constraint on $\mathbf{B}$ (i.e., $\mathbf{B}\mathbf{B}^\top = \mathbf{I}$) is to make the subspace non-redundant [3,11]. Other studies that follow this line of formulation include [3,11,1,95,96].

3 MTL for incomplete data

The discussions so far are only limited to MTL algorithms that use complete data. However, in many applications involving multimodal or longitudinal data, the dataset could be incomplete, i.e., some of the modalities or data at certain time points could be missing. In such cases, we can either 1) use only samples with complete data for MTL study, with the cost of reduced statistical power of analysis due to smaller dataset, or 2) impute the missing data before performing the MTL study, where the imputation is very much prone to error for data missing in blocks, or 3) design a MTL method that is applicable to incomplete data, such as [92,88].

In MTL for incomplete data [92,88], the prediction of one modality combination is treated as one task, and the aim is to learn multiple tasks simultaneously. Specifically, this method first groups the incomplete dataset into several subsets of complete data, each subset of data is comprised of different combination of modalities. Then, treating the prediction or classification using each data subset as task, the models in [92,88] simultaneously learn all the tasks jointly. This method avoids the drawback of data imputation while allows the use of all available data for analysis [74].

Figure 2 shows a typical multi-task learning (MTL) framework for dealing with classification problems involving incomplete multimodal data [92]. The formulation for this framework is given as:

$$\min_{\mathbf{W}=[\mathbf{W}^1\mathbf{W}^2\cdots\mathbf{W}^m]} \frac{1}{m}\sum_{i=1}^{m}\frac{1}{N^i}L(\mathbf{X}^i,\mathbf{y}^i,\mathbf{W}^i)+\lambda\|\mathbf{W}\|_{2,1} \ , \tag{25}$$

where N^i, $\mathbf{X}^i$, $\mathbf{y}^i$ and $\mathbf{W}^i$ denote the number of samples, the subset data, the target, and the model parameters for the i-th modality combination, respectively, λ is regularization parameter, while $L(\cdot)$ is the loss function for the data fitting term, e.g., logistic loss function (i.e., $\sum_j^{N_i}\ln(1+\exp(-y_j^i(\beta^i)^\top)x_j^i))$ or least square loss function (i.e., $\|\mathbf{X}^i\mathbf{W}^i-\mathbf{y}^i\|_F^2$). Note that the missing data part of $\mathbf{X}^i$ is filled with zeros, while the parameter values in $\mathbf{W}^i$ that corresponding to the missing features in $\mathbf{X}^i$ are fixed at zeros, as shown in Fig. 2. The formulation given in Eq. 25 is a simplified equivalent version of the formulation given in [92] for better comprehension. Xiang *et al.* [88] has extended this formulation by allowing overlap of samples in each modality combination, and introducing extra regularizing parameters for each modality in the data fitting term, as shown below

$$\min_{\mathbf{W},\boldsymbol{\alpha}} \frac{1}{m}\sum_{i=1}^{m}\frac{1}{N^i}\ L(\mathbf{X}^i,\mathbf{y}^i,\mathbf{W},\boldsymbol{\alpha}^i)+\lambda R_\beta(\mathbf{W}), \quad s.t.\ \ R_\alpha(\boldsymbol{\alpha}^i)<1 \ , \tag{26}$$

where $\boldsymbol{\alpha}^i=[\alpha^{i,1}\alpha^{i,2}\cdots\alpha^{i,S}]$ is a weight vector for i-th modality combination, with each of its element denotes the weight for one modality source (assuming there are a total of S modality sources, the length of $\boldsymbol{\alpha}^i$ would be S). Note that unlike the work in [92], a consistent weight vector $\mathbf{W}$ is learned for all the modality combinations, and the different weight vector for each task (recall that a task is a prediction using samples from a certain modality combination) is achieved by the combination of $\mathbf{W}$ and $\boldsymbol{\alpha}^i$. For instance, for a least square loss, we have $L(\cdot)=\|\mathbf{y}^i-\sum_{j=1}^{S}\alpha^{i,j}\mathbf{X}^{i,j}\mathbf{W}^j\|_F^2$, where $\mathbf{X}^{i,j}$ denotes the j-th modality data of the i-th modality combination, $\alpha^{i,j}$ denotes the modality

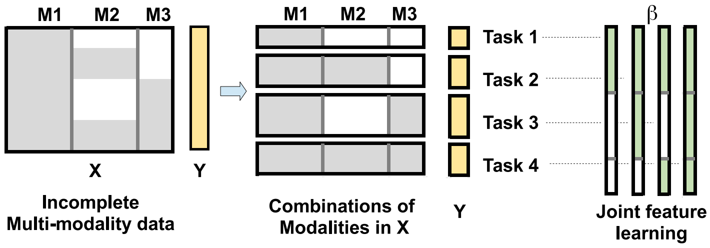


Fig. 2: Left: Original classification problem with incomplete multimodal data. Right: Multi-task learning for incomplete multimodal data [92]. (X: Input data; Y: Target data; M1, M2, M3: Multimodal data; White: Missing data; Grey: Available data, Green: Learned weights)

source weight for this part of the data (if that data source is missing, then its value is zero, otherwise, the weights are learned in formulation Eq. (26)), while $\mathbf{W}^j$ is the weight vector for the j-th modality. On the other hand, R_β and R_α are the regularizers for the weight vector $\mathbf{W}$ and $\boldsymbol{\alpha}^i$, respectively.

4 MTL for deep learning

The formulations we discussed so far use hand crafted features and assume there are linear relationships between the data and target labels. However, in many applications where a complex data-to-target relationship exists [74], this too conservative assumption may limit the performance of prediction models. Recently, thanks to its capability in learning latent representation of the data without any explicit hand-crafted formulation, deep learning technique or deep neural network structure has been adopted for multi-task learning [69]. Most of the applications utilizes either the hard (i.e., sharing the hidden layers between all tasks) or soft (i.e., each task has its own model (hidden layers) with its own parameters) parameter sharing.

Fig. 3 shows a typical example of multi-task deep learning used in the literature [8,71,93,87,104,97,65], in which we assume there is only one input data with multiple output targets. As shown in Fig. 3, there are generally two types of hidden layers in the model, i.e., the shared layers and task-specific layers. The shared layers learn the intrinsic low-level representation of the data, which are general among all the tasks, while the task-specific layer learns the classification network parameters that map the learned latent representations from the previous shared layers to the task-specific output layers.

The combination of multi-task learning and deep learning has been employed for computer vision [21,20,66,42]. As mentioned previously, the multi-task model performs well only if the jointly learned tasks are related among each other. However, there is no adequate definition of task relevance or task-relatedness, leading to the difficulty of grouping the related tasks for joint learning. To overcome this issue, Fang *et al.,* [21] proposed a dynamic multi-task convolutional neural network (DMT CNN) with each task holds a subnet

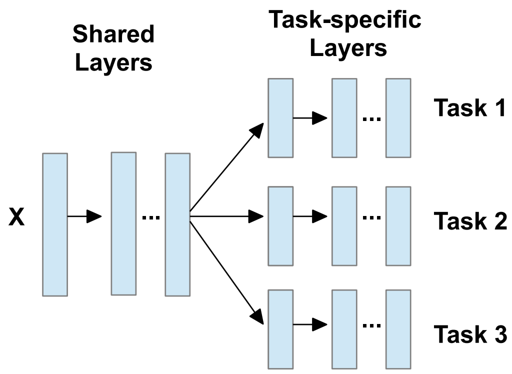


Fig. 3: Multi-task learning for deep learning

and the degree of information shared among subnets (or tasks) is flexible. This is achieved by including dynamic connections called *task transfer connections* among subnets that are learned dynamically to measure the impact of supervisory signals (e.g., the class label) from the higher layers on a certain lower layer, leading to a automatic grouping of tasks during training based on the degree of relevance among tasks. One additional advantage of this model is its incremental learning capability where subnet of a new task can be augmented to the existing learned model without the need of updating the already learned parameters.

Ranjan *et al.,* [66] proposed a deep multi-task learning framework that learns face detection, landmark localization, pose estimation and gender recognition simultaneously from images. In this framework, intermediate layers of a deep CNN are fused using a separate CNN followed by a multi-task learning that operates on the fused features which are based on the synergy among tasks [66].

On the other hand, Fan *et al.,* [20] proposed a deep multi-task learning algorithm for large-scale visual recognition that can recognize more than ten thousands of object classes. Multiple sets of deep features are first extracted from the different layers of a deep CNN before the visually-similar object classes are assigned to the same group. Based on the inter-task relatedness or similarities, more discriminative group-specific deep representations can be learned jointly with a more discriminative tree classifier. This algorithm is able to learn new object class when given a new training image through a incremental deep learning approach.

Besides applications in the more conventional computer vision field, the deep multi-task learning is also applied in medical image analysis recently [81, 57, 70, 90, 9, 23, 101, 2, 50, 98]. Wachinger *et al.,* [81] designed a 3D CNN based segmentation algorithm to segment neuroanatomy from T_1-weighted magnetic resonance images by jointly learning an abstract feature representation and a multi-class classification. The algorithm, named as DeepNAT, predicts not only the center voxel of the patch but also its neighbors via multi-task learning. In another medical image analysis study, Moeskops *et al.,* [57] proposed a deep learning algorithm that based on CNN for different segmentation tasks including brain MRI, breast MRI and cardiac CT angiography (CTA). This deep multi-task algorithm is able to identify the imaging modality, the visualized anatomical structures, and the tissue classes simultaneously. It also demonstrates potential application of a single system in clinical practice to automatically perform diverse segmentation tasks without task-specific training [57].

There is a special type of deep neural network where no shared layers are used, i.e., the network parameters are not shared among different tasks directly [15]. Rather, the information among different tasks is shared by imposing regularization (e.g., look-up table) between network parameters of different tasks.

4.1 Resources and tools

The MTL algorithms have been implemented in various platforms. For Matlab user, the available MTL tools include Multi-Task Learning via StructurAl Regularization (MALSAR) [107], Matlab MTL [14,13,6,33,103,99], Argyrious's implementation of MTL [4,7,5], etc.

Online resources for neural network based MTL include [25], which describes how to use tensorflow to train a multi-task learning neural network, [69] which gives a very comprehensive overview of MTL in deep learning, [16], which releases python code for neural network MTL, [54], which describes how to implement MTL in Keras, etc.

5 Conclusion

In this paper, we have briefly discussed the motivation of multi-task learning, review some of the current multi-task learning (MTL) methods, and compare their formulations. In addition, we also discuss how MTL can be used for incomplete multi-modality or longitudinal data (i.e., some modalities or data at certain time points are missing), how neural network or deep learning can be used to solve complicated multi-task learning problems, and some of the useful resources and tools that can be used to help us implement codes to solve MTL problem. We hope that this review can be served as guidance for general readers that are new in this area to have a quick understanding of the key idea in multi-task learning, and as a brief refresh for the more advanced users on this topic.

References

1. Agarwal, A., Gerber, S., Daume, H.: Learning multiple tasks using manifold regularization. In: Advances in neural information processing systems. pp. 46–54 (2010)

2. Ahmed, B., Thesen, T., Blackmon, K., Kuzniecky, R., Devinsky, O., Dy, J., Brodley, C.: Multi-task learning with weak class labels: Leveraging ieeg to detect cortical lesions in cryptogenic epilepsy. In: Machine Learning for Healthcare Conference. pp. 115–133 (2016)
3. Ando, R.K., Zhang, T.: A framework for learning predictive structures from multiple tasks and unlabeled data. Journal of Machine Learning Research 6(Nov), 1817–1853 (2005)
4. Argyriou, A.: Machine learning software. http://ttic.uchicago.edu/ argyriou/code/ (2015)
5. Argyriou, A., Evgeniou, T., Pontil, M.: Multi-task feature learning. In: Advances in neural information processing systems. vol. 19, pp. 41–48. MIT Press (2007)
6. Argyriou, A., Evgeniou, T., Pontil, M.: Convex multi-task feature learning. Machine Learning 73(3), 243–272 (2008)
7. Argyriou, A., Micchelli, C.A., Pontil, M., Ying, Y.: A spectral regularization framework for multi-task structure learning, nips 20. Journal Publications on Mathematics (Harmonic Analysis) (2008)
8. Caruana, R.: Multitask learning. In: Learning to learn, pp. 95–133. Springer (1998)
9. Chaichulee, S., Villarroel, M., Jorge, J., Arteta, C., Green, G., McCormick, K., Zisserman, A., Tarassenko, L.: Multi-task convolutional neural network for patient detection and skin segmentation in continuous non-contact vital sign monitoring. In: 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017). p. 5110 (2017)
10. Chen, J., Liu, J., Ye, J.: Learning incoherent sparse and low-rank patterns from multiple tasks. ACM Trans. Knowl. Discov. Data 5(4), 22:1–22:31 (2012)
11. Chen, J., Tang, L., Liu, J., Ye, J.: A convex formulation for learning shared structures from multiple tasks. In: Proceedings of the 26th Annual International Conference on Machine Learning. pp. 137–144. ACM (2009)
12. Chen, J., Zhou, J., Ye, J.: 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. pp. 42–50. ACM (2011)
13. Ciliberto, C.: Matmtl. https://github.com/cciliber/matMTL (2017)
14. Ciliberto, C., Mroueh, Y., Poggio, T.: Convex learning of multiple tasks and their structure. In: International Conference on Machine Learning (ICML) (2015)
15. Collobert, R., Weston, J.: A unified architecture for natural language processing: Deep neural networks with multitask learning. In: Proceedings of the 25th international conference on Machine learning. pp. 160–167. ACM (2008)
16. Crichton, G., Pyysalo, S.: Code supporting: A neural network multi- task learning approach to biomedical named entity recognition. https://doi.org/10.17863/CAM.12584 (2017), software
17. Elgammal, A., Lee, C.S.: Separating style and content on a nonlinear manifold. In: Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. vol. 1, pp. I–I. IEEE (2004)
18. Evgeniou, T., Micchelli, C.A., Pontil, M.: Learning multiple tasks with kernel methods. Journal of Machine Learning Research 6(Apr), 615–637 (2005)
19. Evgeniou, T., Pontil, M.: Regularized multi–task learning. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 109–117. ACM (2004)
20. Fan, J., Zhao, T., Kuang, Z., Zheng, Y., Zhang, J., Yu, J., and, J.P.: Hd-mtl: Hierarchical deep multi-task learning for large-scale visual recognition. IEEE Transactions on Image Processing 26(4), 1923 – 1938 (2017)
21. Fang, Y., Ma, Z., Zhang, Z., Zhang, X.Y., Bai, X.: Dynamic multi-task learning with convolutional neural network. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17. pp. 1668–1674 (2017), https://doi.org/10.24963/ijcai.2017/231
22. Fazel, M.: Matrix rank minimization with applications. Ph.D. thesis, Department of Electrical Engineering, Stanford University (2002)
23. Ghafoorian, M., Karssemeijer, N., Heskes, T., van Uden, I.W.M., Sanchez, C.I., Litjens, G., de Leeuw, F.E., van Ginneken, B., Marchiori, E., Platel, B.: Location sensitive

deep convolutional neural networks for segmentation of white matter hyperintensities. Scientific Reports 7(1), 5110– (2017), https://doi.org/10.1038/s41598-017-05300-5
24. Girshick, R.: Fast r-cnn. In: IEEE International Conference on Computer Vision. p. 1440 1448 (2015)
25. Godwin, J.: Multi-task learning in tensorflow: Part 1. https://www.kdnuggets.com/2016/07/multi-task-learning-tensorflow-part-1.html (2018)
26. Gong, P., Ye, J., Zhang, C.s.: Multi-stage multi-task feature learning. In: Advances in Neural Information Processing Systems. pp. 1988–1996 (2012)
27. Gong, P., Ye, J., Zhang, C.: Robust multi-task feature learning. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 895–903. ACM (2012)
28. Gong, P., Zhou, J., Fan, W., Ye, J.: Efficient multi-task feature learning with calibration. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 761–770. ACM (2014)
29. Han, L., Zhang, Y.: Learning tree structure in multi-task learning. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 397–406. ACM (2015)
30. Han, L., Zhang, Y.: Multi-stage multi-task learning with reduced rank. In: AAAI. pp. 1638–1644 (2016)
31. Han, L., Zhang, Y., Song, G., Xie, K.: Encoding tree sparsity in multi-task learning: A probabilistic framework. In: AAAI. pp. 1854–1860 (2014)
32. Hu, R., Zhu, X., Cheng, D., He, W., Yan, Y., Song, J., Zhang, S.: Graph self-representation method for unsupervised feature selection. Neurocomputing 220, 130–137 (2017)
33. Jacob, L., Vert, J.p., Bach, F.R.: Clustered multi-task learning: A convex formulation. In: Advances in neural information processing systems. pp. 745–752 (2009)
34. Jalali, A., Ravikumar, P., Sanghavi, S.: A dirty model for multiple sparse regression. IEEE Transactions on Information Theory 59(12), 7947–7968 (2013)
35. Jalali, A., Sanghavi, S., Ruan, C., Ravikumar, P.K.: A dirty model for multi-task learning. In: Lafferty, J.D., Williams, C.K.I., Shawe-Taylor, J., Zemel, R.S., Culotta, A. (eds.) Advances in Neural Information Processing Systems 23, pp. 964–972. Curran Associates, Inc. (2010)
36. Jebara, T.: Multi-task feature and kernel selection for svms. In: Proceedings of the twenty-first international conference on Machine learning. p. 55. ACM (2004)
37. Jebara, T.: Multitask sparsity via maximum entropy discrimination. Journal of Machine Learning Research 12(Jan), 75–110 (2011)
38. Kim, S., Xing, E.P.: Tree-guided group lasso for multi-task regression with structured sparsity. In: International Conference on International Conference on Machine Learning. pp. 543–550 (Jun 2010)
39. Lee, H., Battle, A., Raina, R., Ng, A.Y.: Efficient sparse coding algorithms. In: Advances in neural information processing systems. pp. 801–808 (2007)
40. Lee, S., Zhu, J., Xing, E.P.: Adaptive multi-task lasso: with application to eqtl detection. In: Advances in neural information processing systems. pp. 1306–1314 (2010)
41. Li, C., Gupta, S., Rana, S., Nguyen, V., Venkatesh, S., Ashley, D., Livingston, T.: Multiple adverse effects prediction in longitudinal cancer treatment. In: Pattern Recognition (ICPR), 2016 23rd International Conference on. pp. 3156–3161. IEEE (2016)
42. Li, X., Zhao, L., Wei, L., Yang, M.H., Wu, F., Zhuang, Y., Ling, H., and, J.W.: Deepsaliency: Multi-task deep neural network model for salient object detection. IEEE Transactions on Image Processing 25(8), 3919 – 3930 (2016)
43. Liu, F., Wee, C.Y., Chen, H., Shen, D.: Inter-modality relationship constrained multi-modality multi-task feature selection for alzheimer's disease and mild cognitive impairment identification. NeuroImage 84, 466–475 (2014)
44. Liu, G., Yan, Y., Song, J., Sebe, N.: Minimizing dataset bias: Discriminative multi-task sparse coding through shared subspace learning for image classification. In: Image Processing (ICIP), 2014 IEEE International Conference on. pp. 2869–2873. IEEE (2014)
45. Liu, H., Palatucci, M., Zhang, J.: 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. pp. 649–656. ACM (2009)

46. Liu, J., et al.: SLEP: Sparse Learning with Efficient Projections. Arizona State University (2009)
47. Liu, J., Ji, S., Ye, J.: Multi-task feature learning via efficient l 2, 1-norm minimization. In: Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence. pp. 339–348. AUAI Press (2009)
48. Liu, J., Ye, J.: Efficient euclidean projections in linear time. In: Proceedings of the 26th Annual International Conference on Machine Learning. pp. 657–664. ACM (2009)
49. Liu, J., Ye, J.: Moreau-yosida regularization for grouped tree structure learning. In: Advances in Neural Information Processing Systems. pp. 1459–1467 (2010)
50. Liu, M., Zhang, J., Adeli, E., Shen, D.: Deep multi-task multi-channel learning for joint classification and regression of brain status. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 3–11. Springer (2017)
51. Lounici, K., Pontil, M., Tsybakov, A.B., Van De Geer, S.: Taking advantage of sparsity in multi-task learning. arXiv preprint arXiv:0903.1468 (2009)
52. Lozano, A.C., Swirszcz, G.: Multi-level lasso for sparse multi-task regression. In: Proceedings of the 29th International Coference on International Conference on Machine Learning. pp. 595–602. Omnipress (2012)
53. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online dictionary learning for sparse coding. In: Proceedings of the 26th annual international conference on machine learning. pp. 689–696. ACM (2009)
54. Mandal, M.K.: Multi-task learning in keras — implementation of multi-task classification loss. https://blog.manash.me/multi-task-learning-in-keras-implementation-of-multi-task-classification-loss-f1d42da5c3f6 (Feb 2018)
55. Maurer, A., Pontil, M., Romera-Paredes, B.: Sparse coding for multitask and transfer learning. In: International Conference on Machine Learning. pp. 343–351 (2013)
56. McDonald, A.M., Pontil, M., Stamos, D.: Spectral k-support norm regularization. In: Advances in Neural Information Processing Systems. pp. 3644–3652 (2014)
57. Moeskops, P., Wolterink, J.M., van der Velden, B.H.M., Gilhuijs, K.G.A., Leiner, T., Viergever, M.A., Isgum, I.: Deep learning for multi-task medical image segmentation in multiple modalities. CoRR abs/1704.03379 (2017), http://arxiv.org/abs/1704.03379
58. Negahban, S., Wainwright, M.J.: Joint support recovery under high-dimensional scaling: Benefits and perils of $\ell_{1,\infty}$-regularization. In: Proceedings of the 21st International Conference on Neural Information Processing Systems. pp. 1161–1168. Curran Associates Inc. (2008)
59. Ng, A.: Multi-task learning. https://www.coursera.org/learn/machine-learning-projects/lecture/l9zia/multi-task-learning (May 2018)
60. Obozinski, G., Taskar, B., Jordan, M.: Multi-task feature selection. Statistics Department, UC Berkeley, Tech. Rep 2 (2006)
61. Obozinski, G., Taskar, B., Jordan, M.I.: Joint covariate selection and joint subspace selection for multiple classification problems. Statistics and Computing 20(2), 231–252 (2010)
62. Olshausen, B.A., Field, D.J.: Sparse coding with an overcomplete basis set: A strategy employed by v1? Vision research 37(23), 3311–3325 (1997)
63. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on knowledge and data engineering 22(10), 1345–1359 (2010)
64. Pong, T.K., Tseng, P., Ji, S., Ye, J.: Trace norm regularization: Reformulations, algorithms, and multi-task learning. SIAM Journal on Optimization 20(6), 3465–3489 (2010)
65. Ranjan, R., Patel, V.M., Chellappa, R.: Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)
66. Ranjan, R., Patel, V.M., Chellappa, R.: Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)

67. Rao, N., Cox, C., Nowak, R., Rogers, T.T.: Sparse overlapping sets lasso for multi-task learning and its application to fmri analysis. In: Advances in neural information processing systems. pp. 2202–2210 (2013)
68. Romera-Paredes, B., Argyriou, A., Berthouze, N., Pontil, M.: Exploiting unrelated tasks in multi-task learning. In: International Conference on Artificial Intelligence and Statistics. pp. 951–959 (2012)
69. Ruder, S.: An overview of multi-task learning in deep neural networks. arXiv preprint arXiv:1706.05098 (2017)
70. Samala, R.K., Chan, H.P., Hadjiiski, L., Helvie, M.A., Richter, C., Cha, K.: Cross-domain and multi-task transfer learning of deep convolutional neural network for breast cancer diagnosis in digital breast tomosynthesis. In: MICCAI. vol. 10575 (2018), https://doi.org/10.1117/12.2293412
71. Seltzer, M.L., Droppo, J.: Multi-task learning in deep neural networks for improved phoneme recognition. In: Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. pp. 6965–6969. IEEE (2013)
72. Seraj, R.M.: Multi-task learning. Internet: https://www.cs.ubc.ca/~schmidtm/MLRG/Multi-task%20Learning.pdf (2014)
73. Suo, Y., Dao, M., Tran, T., Mousavi, H., Srinivas, U., Monga, V.: Group structured dirty dictionary learning for classification. In: Image Processing (ICIP), 2014 IEEE International Conference on. pp. 150–154. IEEE (2014)
74. Thung, K.H., et al.: Neurodegenerative disease diagnosis using incomplete multi-modality data via matrix shrinkage and completion. Neuroimage 91, 386–400 (2014)
75. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological) pp. 267–288 (1996)
76. Titsias, M.K., Lázaro-Gredilla, M.: Spike and slab variational inference for multi-task and multiple kernel learning. In: Advances in neural information processing systems. pp. 2339–2347 (2011)
77. Turlach, B.A., Venables, W.N., Wright, S.J.: Simultaneous variable selection. Technometrics 47(3), 349–363 (2005)
78. Vasilescu, M.A.O., Terzopoulos, D.: Multilinear image analysis for facial recognition. In: Pattern Recognition, 2002. Proceedings. 16th International Conference on. vol. 2, pp. 511–514. IEEE (2002)
79. Vogt, J., Roth, V.: A complete analysis of the l_1, p group-lasso. arXiv preprint arXiv:1206.4632 (2012)
80. Vounou, M., Nichols, T.E., Montana, G., Initiative, A.D.N., et al.: Discovering genetic associations with high-dimensional neuroimaging phenotypes: a sparse reduced-rank regression approach. Neuroimage 53(3), 1147–1159 (2010)
81. Wachinger, C., Reuter, M., Klein, T.: Deepnat: Deep convolutional neural network for segmenting neuroanatomy. NeuroImage 170, 434–445 (2018), http://www.sciencedirect.com/science/article/pii/S1053811917301465
82. Wang, H., et al.: Facial expression decomposition. In: Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on. pp. 958–965. IEEE (2003)
83. Wang, H., Nie, F., Huang, H., Yan, J., Kim, S., Risacher, S., Saykin, A., Shen, L.: High-order multi-task feature learning to identify longitudinal phenotypic markers for alzheimer's disease progression prediction. In: Advances in Neural Information Processing Systems. pp. 1277–1285 (2012)
84. Wang, J., Ye, J.: Safe screening for multi-task feature learning with multiple data matrices. In: International Conference on Machine Learning. pp. 1747–1756 (2015)
85. Wang, Z., Zhu, X., Adeli, E., Zhu, Y., Nie, F., Munsell, B., Wu, G.: Multi-modal classification of neurodegenerative disease by progressive graph-based transductive learning. Medical image analysis 39, 218–230 (2017)
86. Weiss, K., Khoshgoftaar, T.M., Wang, D.: A survey of transfer learning. Journal of Big Data 3(1), 9 (2016)
87. Wu, Z., Valentini-Botinhao, C., Watts, O., King, S.: Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis. In: Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on. pp. 4460–4464. IEEE (2015)

88. Xiang, S., Yuan, L., Fan, W., Wang, Y., Thompson, P.M., Ye, J., Initiative, A.D.N., et al.: Bi-level multi-source learning for heterogeneous block-wise missing data. NeuroImage 102, 192–206 (2014)
89. Xin, B., Kawahara, Y., Wang, Y., Hu, L., Gao, W.: Efficient generalized fused lasso and its applications. ACM Transactions on Intelligent Systems and Technology (TIST) 7(4), 60 (2016)
90. Xue, W., Brahm, G., Pandey, S., Leung, S., Li, S.: Full left ventricle quantification via deep multitask relationships learning. Medical Image Analysis 43, 54–65 (Jan 2018), `http://dx.doi.org/10.1016/j.media.2017.09.005`
91. Yan, K., Zhang, D., Xu, Y.: Correcting instrumental variation and time-varying drift using parallel and serial multitask learning. IEEE Transactions on Instrumentation and Measurement 66(9), 2306–2316 (2017)
92. Yuan, L., et al.: Multi-source feature learning for joint analysis of incomplete multiple heterogeneous neuroimaging data. NeuroImage 61(3), 622 – 632 (2012)
93. Zhang, C., Zhang, Z.: Improving multiview face detection with multi-task deep convolutional neural networks. In: Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on. pp. 1036–1041. IEEE (2014)
94. Zhang, D., et al.: Multi-modal multi-task learning for joint prediction of multiple regression and classification variables in Alzheimer's disease. Neuroimage 59(2), 895–907 (2012)
95. Zhang, J., Ghahramani, Z., Yang, Y.: Learning multiple related tasks using latent independent component analysis. In: Advances in neural information processing systems. pp. 1585–1592 (2006)
96. Zhang, J., Ghahramani, Z., Yang, Y.: Flexible latent variable models for multi-task learning. Machine Learning 73(3), 221–242 (2008)
97. Zhang, J., Liang, J., Hu, H.: Multi-view texture classification using hierarchical synthetic images. Multimedia Tools and Applications 76(16), 17511–17523 (2017)
98. Zhang, J., Liu, M., Shen, D.: Detecting anatomical landmarks from limited medical imaging data using two-stage task-oriented deep neural networks. IEEE Transactions on Image Processing 26(10), 4753–4764 (2017)
99. Zhang, J., Liu, M., Wang, L., Chen, S., Yuan, P., Li, J., Shen, S.G.F., Tang, Z., Chen, K.C., Xia, J.J., et al.: Joint craniomaxillofacial bone segmentation and landmark digitization by context-guided fully convolutional networks. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 720–728. Springer (2017)
100. Zhang, S., Li, X., Zong, M., Zhu, X., Wang, R.: Efficient knn classification with different numbers of nearest neighbors. IEEE transactions on neural networks and learning systems (2017)
101. Zhang, W., Li, R., Zeng, T., Sun, Q., Kumar, S., Ye, J., Ji, S.: Deep model based transfer and multi-task learning for biological image analysis. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1475–1484. KDD '15, ACM, New York, NY, USA (2015), `http://doi.acm.org/10.1145/2783258.2783304`
102. Zhang, Y., Yang, Q.: A survey on multi-task learning. arXiv preprint arXiv:1707.08114 (2017)
103. Zhang, Y., Yeung, D.Y.: A convex formulation for learning task relationships in multi-task learning. arXiv preprint arXiv:1203.3536 (2012)
104. Zhang, Z., Luo, P., Loy, C.C., Tang, X.: Facial landmark detection by deep multi-task learning. In: European Conference on Computer Vision. pp. 94–108. Springer (2014)
105. Zheng, J., Ni, L.M.: Time-dependent trajectory regression on road networks via multi-task learning. In: AAAI (2013)
106. Zheng, W., Zhu, X., Zhu, Y., Hu, R., Lei, C.: Dynamic graph learning for spectral feature selection. Multimedia Tools and Applications pp. 1–17 (2017)
107. Zhou, J., Chen, J., Ye, J.: Malsar: Multi-task learning via structural regularization. Arizona State University 21 (2011)
108. Zhou, J., Liu, J., Narayan, V.A., Ye, J.: Modeling disease progression via fused sparse group lasso. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1095–1103. ACM (2012)

109. Zhou, J., Yuan, L., Liu, J., Ye, J.: A multi-task learning formulation for predicting disease progression. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 814–822. ACM (2011)
110. Zhou, Y., Jin, R., Hoi, S.C.H.: Exclusive lasso for multi-task feature selection. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. pp. 988–995 (2010)
111. Zhu, X., Li, X., Zhang, S., Ju, C., Wu, X.: Robust joint graph sparse coding for unsupervised spectral feature selection. IEEE transactions on neural networks and learning systems 28(6), 1263–1275 (2017)
112. Zhu, X., Li, X., Zhang, S., Xu, Z., Yu, L., Wang, C.: Graph pca hashing for similarity search. IEEE Transactions on Multimedia 19(9), 2033–2044 (2017)
113. Zhu, X., Suk, H.I., Huang, H., Shen, D.: Structured sparse low-rank regression model for brain-wide and genome-wide associations. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 344–352. Springer (2016)
114. Zhu, X., Suk, H.I., Huang, H., Shen, D.: Low-rank graph-regularized structured sparse regression for identifying genetic biomarkers. IEEE Transactions on Big Data 3(4), 405–414 (2017)
115. Zhu, X., Suk, H.I., Lee, S.W., Shen, D.: Subspace regularized sparse multitask learning for multiclass neurodegenerative disease identification. IEEE Transactions on Biomedical Engineering 63(3), 607–618 (2016)
116. Zhu, X., Zhang, S., Hu, R., Zhu, Y., et al.: Local and global structure preservation for robust unsupervised spectral feature selection. IEEE Transactions on Knowledge and Data Engineering (2017)
117. Zhu, Y., Kim, M., Zhu, X., Yan, J., Kaufer, D., Wu, G.: Personalized diagnosis for alzheimers disease. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 205–213. Springer (2017)
118. Zhu, Y., Zhu, X., Zhang, H., Gao, W., Shen, D., Wu, G.: Reveal consistent spatial-temporal patterns from dynamic functional connectivity for autism spectrum disorder identification. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 106–114. Springer (2016)