Neural Processing Letters (2022) 54:2491–2508
https://doi.org/10.1007/s11063-021-10703-7

# Supervised Shallow Multi-task Learning: Analysis of Methods

**Stanley Ebhohimhen Abhadiomhen[1,2] · Royransom Chimela Nzeh[2] · Ernest Domanaanmwi Ganaa[1,3] · Honour Chika Nwagwu[2] · George Emeka Okereke[2] · Sidheswar Routray[4]**

Accepted: 19 November 2021 / Published online: 29 January 2022


**Abstract**
The last decade has witnessed a continuous boom in the application of machine learning techniques in pattern recognition, with much more focus on single-task learning models. However, the increasing amount of multimedia data in the real world also suggests that these single-task learning models have become unsuitable for complex problems. Hence, multi-task learning (MTL), which leverages the common path shared between related tasks to improve a specific model's performance, has grown popular in the last years. And several studies have been conducted to find a robust MTL method either in the supervised learning or unsupervised learning paradigm using a shallow or deep approach. This paper provides an analysis of supervised shallow-based multi-task learning methods. To begin, we present a rationale for MTL with a basic example that is easy to understand. Next, we formulate a supervised MTL problem to describe the various methods utilized to learn task relationships. We also present an overview of deep learning methods for supervised MTL to compare shallow to non-shallow approaches. Then, we highlight the challenges and future research opportunities of supervised MTL.

**Keywords** Multi-task learning · Supervised learning · Shallow algorithms

✉ Honour Chika Nwagwu
honour.nwagwu@unn.edu.ng

✉ Sidheswar Routray
sidheswar69@gmail.com

Stanley Ebhohimhen Abhadiomhen
5103190343@stmail.ujs.edu.cn ; stanley.abhadiomhen@unn.edu.ng

1 School of Computer Science and Communication Engineering, JiangSu University, Zhenjiang 212013, JiangSu, China

2 Department of Computer Science, University of Nigeria, Nsukka, Nigeria

3 School of Applied Science and Technology, Wa Technical University, Wa Box 553, Ghana

4 Department of Computer Science and Engineering, School of Engineering, Indrashil University, Rajpur, Mehsana, Gujarat, India

## 1 Introduction

Multi-task learning has received enormous interest over the years because it leverages common information between related tasks to improve generalization performance [71]. Although previous efforts were made in the single-task learning setting [41,48,68,72,74,75,86], many studies which are not limited to [5,10,25,26,51,58,60,64,71,78,81,85,94] have also shown that MTL can provide robust improvement to the single-task learning methods with usefulness in applications such as computer vision [60,62,64,92], bioinformatics [19,49] and web search ranking [12,77]. Besides, MTL is related to some sub-fields of machine learning, such as transfer learning [6,27], multi-label learning [33,34] and multi-class learning [17]. However, MTL differs mainly from transfer learning because it learns many related tasks simultaneously to extract shared information.

Therefore, the strategy adopted in MTL as described in [11] is to assume that tasks are often related and may well be unrelated too because a pairwise relationship may exist between several tasks. For example, task $A$ may be related to task $B$ and task $D$ while task $C$ is only related to task $D$. Consequently, the relationships between all or subsets of tasks can be learnt through existing MTL approaches which can be categorized as follows: Regularization based methods [4,5,10,12,19,25,26,40,49,51,58,64,81,85,89,94], low-rank methods [3,29,43,59,67], clustering methods [7,36,44,76], tasks similarity learning methods [22,62,71,90–92] and decomposition methods [37–39,93,96]. These learning methods are widely applied in a supervised learning paradigm using shallow algorithms such as support vector machine (SVM) [10,60,71], single-layer artificial neural network [5,11] and Bayesian network [90,91].

The above implies that most studies on MTL after [11] focuses on supervised learning with minimal efforts made in unsupervised learning [18,28] and reinforcement learning [65]. Besides, there are only a few attempts made in the deep MTL direction [46,70]. It is not surprising, given that shallow models are still more generalized for handling many real-world problems due to some deep learning limitations. For example, whereas shallow models can perform well with a limited quantity of data, a typical deep learning model would require a huge amount of data to perform better than shallow models. Furthermore, suitable theories that can assist researchers in selecting adequate deep learning tools are scarce. As a result, some researchers are still hesitant to embrace the deep learning paradigm. Thus, this paper shall focus on supervised shallow-based multi-task learning SSMTL methods, in which a task with labeled data sets can be a regression or classification problem. Notwithstanding, an overview of deep learning methods is also provided to compare shallow to non-shallow approaches.

The main contributions of the authors in this paper are summarized as follows:

(1) An up-to-date and simplified overview of MTL with illustrating examples.
(2) A presentation of the progresses made in MTL research through the discussion of its existing approaches.
(3) A formulation of a typical SSMTL method using a general approach and an SVM approach to aid analysis of existing SSMTL methods.
(4) An overview of challenges and future research directions of SSMTL.

This paper is structured as follows. Section 2 presents an overview of MTL while providing answers to many questions bothering on MTL. Next, in Sect. 3, an SSMTL problem with a bit more emphasis on the SVM approach is formulated due to its generalization goal. Afterward, Sect. 4 reviews SSMTL methods. Then Sect. 5 provides an overview of deep learning methods in supervised MTL. Furthermore, Sect. 6 presents a discussion of the challenges and future direction of supervised MTL, while Sect. 7 presents the conclusion.

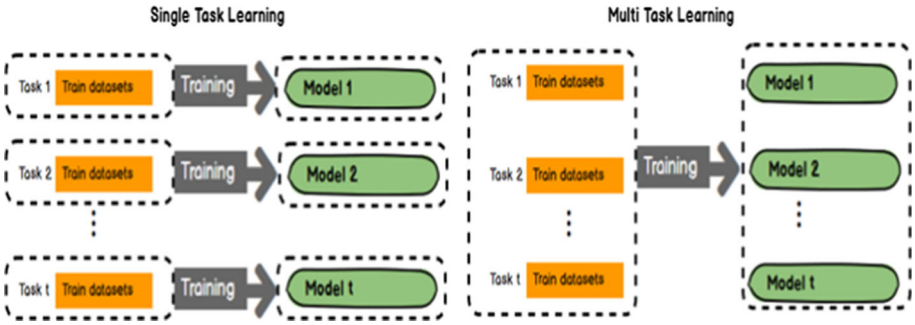


**Fig. 1** An illustration of a single task learning versus multi-task learning

## 2 Why Multi-task Learning?

In order to improve learning accuracy for single-task models, techniques such as ensemble learning [41,74] and transfer learning [6,27] became handy over the years. Specifically, the ensemble technique involves creating multiple models where each model can be combined linearly to produce improved results. On the other hand, the transfer learning technique stores the knowledge gained while learning one task and then use it as a bias to learn another task sequentially. Although both approaches have been substantially demonstrated in the literature to be effective, they both have limitations. For instance, the ensemble technique requires that each model performs better than a random guess to have a favorable output. Otherwise, the worst result may be achieved when compared to a single-task model. This is typical of the transfer learning technique if there is a negative transfer from one task to another. MTL tackles these limitations by providing a way to learn multiple tasks simultaneously to improve performance (see Fig. 1). As such, the MTL process does not just focus on improving prediction accuracy; it also increases data efficiency while reducing training time [73]. For example, through a virtual input, a self-driving vehicle can simultaneously learn the tasks of predicting objects trajectories (avoiding collisions), detecting the location of pedestrians, responding to traffic signals, determining a per-pixel depth using MTL technique with a more reduced training time than in transfer learning. Such that the knowledge gain in one task can be shared simultaneously to learn other tasks, unlike the sequential process of transfer learning, which is more susceptible to negative transfer.

According to the work of [8], MTL is particularly advantageous for learning problems that belong to an environment of related problems. As an example, medical diagnostic challenges are discussed, in which a pathology test can be used to detect numerous diseases at the same time by identifying a common bias that will also aid in learning fresh cases. MTL is also useful in a variety of other complicated real-world situations, like as emotion recognition. In this case, multiple models can be trained at the same time to recognize a dress type and weather condition depending on the learned dress type. Therefore, it's easy to understand how reference [8] and most other preliminary works on MTL, including reference [11] are based on the notion that tasks often share a certain similarity. However, under what condition can one expect different tasks to belong to an environment of related problems? To answer this question, Ben-David and Borbely [9] focuses on sample generating distributions that underpin learning tasks, where task relatedness is defined as an explicit relationship between the distributions. Their idea appears to include a subset of applications where multi-task learning could be effective, while excluding many other MTL scenarios from the picture.

Specifically, the proposed methodology applies to circumstances in which the learner's prior knowledge includes knowledge of some family $F$ of transformations. So, a typical example involves several sensors providing data for the same classification task, such as a system of cameras positioned at the presidential palace's entrance to automatically detect intrusion through the photographs they capture. Thus, let us assume these cameras are placed at different heights, light conditions, and angles. Then it should be clear that each of these cameras has its own bias, which can be difficult to identify. Therefore, Ben-David and Borbely's framework may be utilized to mimic the above in the MTL scenario by developing a collection of picture transformations $F$ so that the data distributions of images collected by all of these cameras are $F$-related.

The illustrations above show MTL problems that cannot be solved well using a single-learning technique. Besides, Liu et al. [52] demonstrate this by assessing individual task performance in MTL and comparing it to single-task learning. Their findings reveal that individual task performance in the MTL context was superior to that of single-task learning, providing a compelling justification for MTL.

## 3 Problem Formulation

Suppose we have $T$ classification tasks, a typical MTL problem formulation using any shallow learning algorithms such as SVM, logistic regression, the artificial neural network can be generalized as follows

### 3.1 General Approach

$$\min_{\mathbf{W}=w_1+w_2\ldots w_t} \sum_{t=1}^{T} \mathcal{L}\left(S_t, w_t\right) + \lambda\left(\mathbf{W}\right), \tag{1}$$

where $\boldsymbol{S_t}$ is the training data for the $t$ task given as follows $\{\boldsymbol{x_{ti}}, y_{ti}\}_{i=1}^{N_t}$ in which $\boldsymbol{x_{ti}} \in R^d$ is the $i$-th training instance of the $t$ task, labeled with $y_{ti}$, $w_t \in R^d$ is the weight vector of the $t$ task, $d$ is the feature space dimension, assuming that each task's input matrix has the same feature dimension (homogeneous feature, but it can alternatively be heterogeneous where $d$ varies per task). The $+$ sign allows $w_1, w_2, w_3 \ldots w_t$ to be concatenated to learn $\boldsymbol{W}$ (i.e., each row of $\boldsymbol{W}$ has a corresponding feature) with a specific regularization constraint denoted by $\lambda(\boldsymbol{W})$ that can be informed mainly by the data's prior knowledge [77]. To illustrate this, we revisit the medical diagnostic and self-driving vehicle problems from Sect. 2, where the input matrices $X_t$ and $X_u$ of two different tasks are the same, but the target outputs $y_t$ and $y_u$ are not. Here, $T$ tasks model can be trained concurrently to learn a common bias for all tasks by carefully selecting the value of $\lambda$( a regularization parameter) to avoid overfitting. It should be emphasized, however, that this does not necessarily indicate a strict MTL problem. As mentioned in reference [88], it may be best be described as a multi-label learning or multi-output problem. As a result, the camera system example also presented in Sect. 2 will more accurately depict a strict MTL problem with different input samples for each task.

### 3.2 Standard SVM Approach

In this section, we employ the SVM MTL formulation from [23] as an example. The reason for this is that SVM is commonly employed to solve MTL problems [66]. Perhaps because

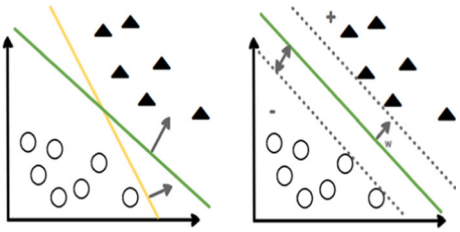

**Fig. 2** The image on the left represents traditional binary classification, while the one on the right side depicts typical SVM binary classification

of its strong generalization capability, which is ideal for MTL. Thus, given training data $\{\{(\boldsymbol{x_i}, y_i)\}_{i=1}^{N}, \boldsymbol{x_i} \in R^d, y_i \in \{1, -1\}\}$, SVM finds the hyperplane with the maximum margin separating the points $-1$ and 1 as illustrated in Fig. 2. As such, the standard soft margin SVM for a single task problem is as follows

$$\begin{aligned} &\min_{w,b} \sum_{i=1}^{N} \xi_i + \lambda \parallel w\parallel^2, \\ &s.t., \ y_i(w.x_i + b) \geq 1 - \xi_i, \ \xi_i \geq 0, \end{aligned} \tag{2}$$

where $\xi_i$ is the slack variable introduced into the constraint to accommodate any outlier that hitherto would not be allowed in the hard margin SVM. In other words, $\xi_i$ measures how much a point violates the margin. $\lambda$ is the regularization constant that regulates the tradeoff between complexity and generalization. It is crucial since any increase in a model's complexity can lead to overfitting, which is when a trained model fits well to train examples but fails to generalize to unknown ones. Therefore, given the same datasets as in Eq. (1), Eq. (16) of [23] provides an example extension of the single task SVM problem of Eq. (2) to MTL SVM as follows

$$\begin{aligned} &\min_{w,B_t} \sum_{t=1}^{T} \sum_{i=1}^{N_t} \xi_{ti} + \lambda \parallel w\parallel^2, \\ &s.t., \ y_{ti} w' B_t x_{ti} \geq 1 - \xi_{ti}, \quad \xi_{ti} \geq 0, \forall t, \forall i, \end{aligned} \tag{3}$$

where the matrix $B_t$ is assumed to be a full rank $d$ for each $t$ to ensure a solution $w$ to the equation exist. Thus, using the Lagrange multiplier approach, a typical solution to Eq. (3) can be found in several steps (the work of [54,80] offers the mathematical deductions of SVM for easy understanding) by first obtaining its dual form. This strategy is beneficial because it incorporates a sparsity effect into SVM by relying on LaGrange multipliers (which are only non-zero at the locations in the margin referred to as support vectors) rather than the feature space, making it computationally efficient for high-dimensional data. Accordingly, the dual problem of Eq. (3) that also considers non-linear cases using the kernel trick (In Reproducing Kernel Hilbert Spaces (RKHSs) [57]) is given in Eq. (18) of [23]. In the next section, we will use the general MTL formulation to review existing SSMTL Methods.

## 4 SSMTL Methods

As shown in Fig. 3, existing SSMTL methods can be categorized into five groups: regularization-based methods, low-rank methods, clustering methods, tasks similarity learning methods, and decomposition methods. We will go through each of these methods in detail in the subsections that follow.

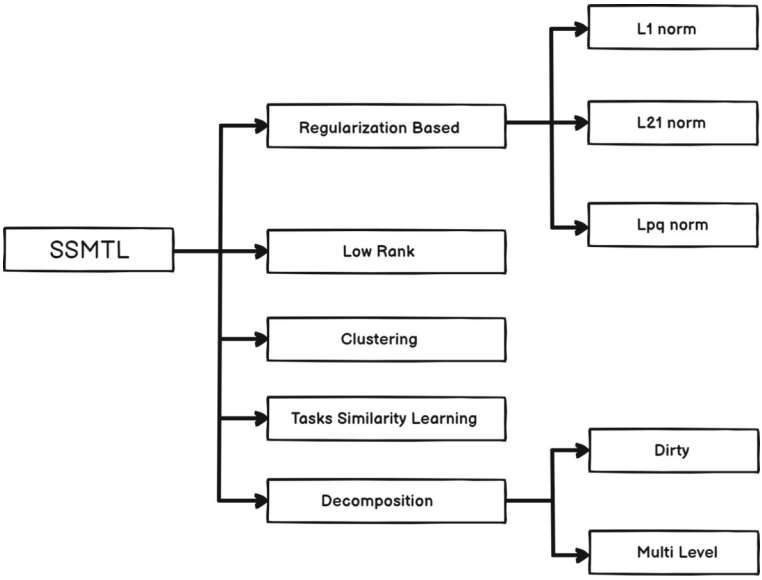


**Fig. 3** Supervised shallow based multitask learning methods

### 4.1 Regularization Based Methods

Regularization concept has been well-known over time, primarily for its application in resolving overfitting and underfitting issues [63] to reduce training and test errors and improve a model's generalization performance. Therefore, various studies [25,27,49] have shown the effectiveness of the regularization technique in MTL, where it is utilized to learn a shared representation across multiple related tasks. For example, if we have a collection of datasets with n correlated features for all tasks, we may use the regularization technique to simultaneously learn an uncorrelated subspace of the original feature space that is shared across all tasks. Particularly in the case of supervised MTL, regularization techniques such as $L_1$-norm, $L_{p,q}$-norm regularization impose a penalty on the weight matrix $\boldsymbol{W}$. This penalty shrinks the row of the weight matrix closer to zero so that only none zero rows are selected. In the next subsections, we will review the existing regularization-based methods.

#### 4.1.1 $L_1$-Norm or Lasso Sparsity

$L_1$-norm which can also be referred to as the least absolute shrinkage and selection operator (Lasso) penalty, is an alternative to $L_2$-norm. Although, the $L_2$-norm can be used minimize computing complexity while boosting performance accuracy by shrinking the rows of the weight matrix closer to zero, it cannot impose sparsity on the weight matrix. As a result, $L_2$-norm cannot perform feature selection automatically. So, to illustrate the capability of $L_1$-norm, we consider the $L_1$-norm version of Eq. (1), which is given as follows

$$\min_{\boldsymbol{W}} \sum_{t=1}^{T} \mathcal{L}\left(S_t, w_t\right) + \lambda \parallel \boldsymbol{W} \parallel_1, \tag{4}$$

It is easy to see that the $L_1$-norm regularization in Eq. (4) is non-differentiable. As such, a large value for the regularization constant $\lambda$ will cause some rows of the weight matrix

whose columns are the $T$ tasks specific weight vector to be exactly zero. This characteristic of $L_1$-norm encourages sparsity of the feature space but it fails to perform group selection in cases where there are several correlated features that are all important in determining the target variable. That is because, when the above is the case, $L_1$-norm will select only a few features while it shrinks the others to zero. In doing so, $L_1$-norm will fail to capture an absolute relationship between the T tasks. Due to this limitation, $L_1$-norm is often applied in combination with other norms. For instance, several variants of $L_1$-norm such as $L_{1,2}$-norm [25,95], $L_{1,\infty}$-norm [15,38], $L_{1,1}$-norm [56] have been used to capture sparse representation shared across tasks. Specifically, Gong et al. [25] proposes a method based on capped-$L_1$, $L_1$ norm as follows

$$\min_{\boldsymbol{W}} \mathcal{L}(\boldsymbol{W}) + \lambda \sum_{j=1}^{d} m(\| w_j \|_1, \theta) : \boldsymbol{W} \in R^{d \times T}. \tag{5}$$

First, in Eq. (5), an $L_1$ norm penalty is imposed on the row of the weight matrix $\boldsymbol{W}$ to obtain a sparse representation for all related tasks. Then, a capped- 1 norm which was initially proposed in [87] is further imposed on the weight matrix. With this combination, the optimal $\boldsymbol{W}$ in Eq. (5) would have many non-zero rows. Besides, it can observe also through Eq. (5) that $\boldsymbol{W}$ is threshold by a parameter $\theta$ ,where $w_j^{1 \times T}$ denotes the $jth$ row of $W$. In other words, the threshold parameter $\theta$ regulates $\boldsymbol{W}$'s sparsity such that as it becomes smaller, the rows of $\boldsymbol{W}$ gets sparser. Thus, making only a subset of features to be utilized. Moreover, the work of [47], which proposed the GO-MTL for Grouping and Overlap Multi-Task Learning, had also previously used the $L_1$-norm to impose sparsity on a matrix $\boldsymbol{S} \in R^{k \times T}$ containing the weights of a linear combination of each task. This approach enforces that each observed task is obtained from only a few of the $k$ latent tasks. Such that, the weight matrix $\boldsymbol{W}$ can be calculated as $\boldsymbol{W} = \boldsymbol{LS}$, where $\boldsymbol{L}$ is a matrix of size $d \times k$ with each column representing a latent task.

#### 4.1.2 $L_{2,1}$-Norm for Group Sparsity

$L_{2,1}$-norm is a variant of $L_2$-norm, which can be applied for group feature selection. This is because $L_{2,1}$-norm can capture tasks relatedness using a shared representation of a similar set of features amongst related tasks. Accordingly, Eq. (4) can be extended for group sparsity based on the $L_{2,1}$-norm as follows

$$\min_{\boldsymbol{W}} \sum_{t=1}^{T} \mathcal{L}(S_t, w_t) + \lambda \parallel \boldsymbol{W} \|_{2,1}, \tag{6}$$

Several MTL research works based on the $L_{2,1}$-norm includes [4,5,19,26,49,51,55,64]. In particular, Argyriou et al. [5] proposed a convex optimization problem based $L_{2,1}$-norm as follows

$$\min_{\boldsymbol{A},\boldsymbol{U}} \sum_{t=1}^{T} \sum_{i=1}^{m} \mathcal{L}(y_{ti}, \langle a_t, \boldsymbol{U}^{\boldsymbol{T}} x_{ti} \rangle) + \gamma \parallel \boldsymbol{A} \|_{2,1}^2 : \boldsymbol{A} \in R^{d \times T}, \tag{7}$$

Basically, Eq. (7) is formed under the assumption that all related tasks share small feature sets with $N \leq d$. It then means that matrix $\boldsymbol{A}$ will have many zero rows, corresponding to the columns of matrix $\boldsymbol{U}$ (the irrelevant features) not required by any task. Therefore, to learn the required features $N$, the $L_{2,1}$-norm regularization is introduced to ensure that matrix $\boldsymbol{A}$ has a small number of non-zero rows. Besides, by removing matrix $\boldsymbol{U}$, it is clear that Eq. (7) is comparable to the method proposed in [64], which uses the $L_{2,1}$-norm to select a subset

of features that is good for all tasks. However, in a single task scenario, this method will reduce to an $L_1$-norm approach. Furthermore, Li et al. [49] applied $L_{2,1}$-norm for survival analysis in MTL scenario by means of a single base kernel. Yet, this single kernel approach was extended to multiple kernels in [19] to demonstrate that survival analysis in MTL can benefit more by capturing a shared representation through more gene data sources. Hence an additional data source (pathways/gene datasets) is incorporated to identify survival-related molecular mechanisms. Such that one kernel is used for the cancer survival benchmark dataset and another kernel for the pathways/gene dataset. This approach, however, does not utilize the $L_{2,1}$-norm as the regularization term. Besides, for $p, q \geq 1$, Eq. (6) can generalize to $L_{p,q}$-norm as follows

$$\min_{\boldsymbol{W}} \sum_{t=1}^{T} \mathcal{L}\left(S_t, w_t\right) + \lambda \parallel \boldsymbol{W} \parallel_{p,q}, \tag{8}$$

where the $L_p$-norm is applied on the rows, followed by the $L_q$-norm on the vector of row norms. Therefore, the variants of $L_{p,q}$-norm include $L_{p,1}$-norm and capped $L_{p,1}$-norm but, $L_{p,1}$-norm is the same as $L_{2,1}$-norm in Eq. (6) if $p = 2$. Thus like Eq. (5), a capped $L_{p,1}$-norm can be obtained as follows

$$\min_{\boldsymbol{W}} \sum_{t=1}^{T} \mathcal{L}\left(S_t, w_t\right) + \lambda \sum_{j=1}^{d} m(\parallel w_j \parallel_p, \theta) : \boldsymbol{W} \in R^{d \times T}. \tag{9}$$

In any case, when the threshold parameter $\theta$ in Eq. (9) becomes too large, the capped $L_{p,1}$-norm will reduce to $L_{p,1}$-norm. To conclude this section, it may be worth mentioning that aside from the norms discussed above, the cluster norm [35] and $K$ support norm [59] can also be utilized to learn a better similarity weight matrix. Moreover, to considerably acquire an accurate similarity between tasks, the Multitask Learning problem was previously formulated as a Multiple Kernel Learning [53] one by Widmer et al. [82] using a q-Norm MKL algorithm. And this approach was shown to outperform similar baseline methods.

### 4.2 Low-Rank

Theoretically, the weight matrix $\boldsymbol{W}$ can be assumed to be low-rank since tasks are usually related with similar model parameters. However, to obtain the low rank of $\boldsymbol{W}$, one can solve the following nuclear norm-based optimization problem:

$$\min_{\boldsymbol{W}} \sum_{t=1}^{T} \mathcal{L}\left(S_t, w_t\right) + \lambda \parallel \boldsymbol{W} \parallel_*, \tag{10}$$

where $\boldsymbol{W} \in R^{d \times T}$ denotes the weight matrix, $\parallel . \parallel_*$ denotes nuclear norm [1,2]. Actually, the methods in [3,14,43,67] use a similar approach as Eq. (10) to find the low-rank representation of $\boldsymbol{W}$. For example, reference [3] proposed a non-convex formulation to learn a low dimensional subspace shared between multiple related tasks under the assumption that all related tasks have similar model parameters. As such, the weight vector of the $t$ task can be obtained as follows

$$w_t = u_t + \Theta^T v_t, \tag{11}$$

where $u_t$ is the t task learned weight vector, $\boldsymbol{\Theta}$ is the low rank representation of $\boldsymbol{W}$ and $v_t$ is the bias for the $t$ task. Since Eq. (10) is non-convex, it will be difficult to solve it, especially when the feature space is highly correlated. Hence, to relax the non-convex approach, Chen

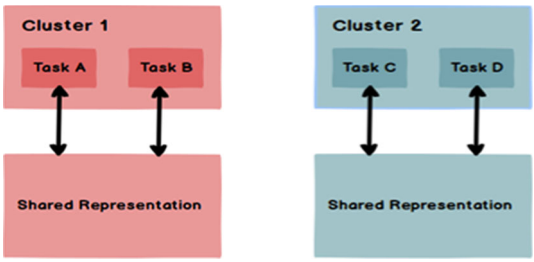


**Fig. 4** Task clustering method

et al. [13] proposed a convex formation that is much easier to solve. It uses $L_{2,2}$-norm (which is a special case when $p = q = 2$ for the $L_{p,q}$-norm in Eq. (8)), also known as the Frobenius norm or the Hilbert–Schmidt norm, to penalize eigenvalues. This approach, however, uses complex constraints, so it is not scalable to large data sets. Furthermore, to learn a better low-rank matrix, [29] extended the idea in [3,67] by introducing a capped trace norm regularization as follows

$$\min_{\boldsymbol{W}} \mathcal{L}(\boldsymbol{W}) + \lambda \sum_{i=1}^{R} min(\sigma_i(\boldsymbol{W}), \tau) : \boldsymbol{W} \in R^{d \times T}, \tag{12}$$

where $\sum_{i=1}^{R} \sigma_i(\boldsymbol{W})$ is denoted as the set of non-increasing ordered singular values of $\boldsymbol{W}$. Noticeably, Eq. (12) is like the capped $L_{p,1}$-norm in Eq. (9) because it can be reduced to Eq. (10) when the threshold parameter $\tau$ becomes very large, e.g., $\tau \rightarrow \infty$. However, these approaches cannot guarantee robust classification results when the data originate from nonlinear subspaces. Therefore, several kernel-based approaches have been proposed over the years which focus on tackling the above issue. For example, to handle multiple features from the variational mode decomposition (VMD) domain, He et al. [31] proposed the kernel low-rank multitask learning (KL-MTL). KL-MTL uses the Low-rank representation (LRR) [50] nuclear norm strategy to capture the global structure of multiple tasks, then using the kernel trick, this approach was extended for nonlinear low-rank multitask learning. Besides, the KL-MTL approach was further expanded in [32] to handle 2-D variational mode decomposition (2-D-VMD). Subsequently, Tian et al. [78] proposed a nonparametric multitask learning method, which measures the task relatedness in a reproducing kernel Hilbert space (RKHS). Specifically, the multitask learning problem is formulated as a linear combination of common eigenfunctions shared by different tasks and individual task's unique eigenfunctions. In this way, each task's eigenfunctions can then provide some additional information to another and so as to improve generalization performance.

### 4.3 Clustering

As we saw in Sect. 1, a pairwise relationship can exist among the tasks, where Task $A$ is only related to Task $B$, and Task $C$ is only related to Task $D$. Thus, the clustering method can be used to learn model parameters by placing all related but separate tasks in the same cluster where they are co-learned. As a result, the work of [7,76] proposed methods, which obtains the model parameters by clustering tasks(see Fig. 4 for illustration) into group of related tasks based on prior knowledge obtained in the single task setting. However, the downside is that not too good model parameters can be learned in this two-stage approach resulting in poor generalization performance for all tasks. To address the above weakness, Kang et al. [44] proposed a method that can determine the pairwise relationship existing between tasks

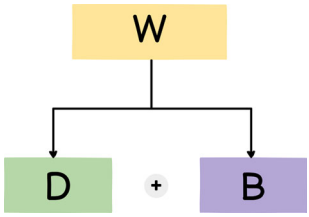


**Fig. 5** Dirty decomposition method

while obtaining their parameter. It is achieved by solving the single optimization problem below.

$$\boldsymbol{W^*} = \min \sum_t \mathcal{L}\left(D_t, w_t\right) + \gamma \sum_g \parallel \boldsymbol{W_g} \parallel_*^2, \tag{13}$$

where $G$ denotes the number of clusters available for all tasks. As a result, the weight matrix of the tasks in the $g$th cluster is denoted by $\boldsymbol{W_g}$. With this formulation, all tasks in the same cluster can be co-learned in contrast to the tasks in other clusters. Thus, $\boldsymbol{W_g}$ can be obtained as follows

$$\parallel \boldsymbol{W}\,\boldsymbol{Q_g} \parallel_* = Trace[\boldsymbol{W}\,\boldsymbol{Q_g}(\boldsymbol{W}\,\boldsymbol{Q_g})^T]^{\frac{1}{2}}, \tag{14}$$

where $\boldsymbol{Q}$ is the group assignment matrix composed of $q_{gt} \in \{0, 1\}$. That is, 0 and 1 indicates whether the $t$ task is assigned to $g$th cluster or not. Then $\boldsymbol{Q_g} \in R^{T \times T}$ is a diagonal matrix with $q_{gt}$ as the diagonal elements. This method is very effective because the fact that tasks are related does not automatically suggest that successful sharing will occur between them. Furthermore, Jacob et al. [36] introduced a new spectral norm that encodes the priori assumption (tasks within a group have similar weight vectors) without prior knowledge of task grouping. This approach was shown to outperform similar state-of-the-arts methods. Subsequently, reference [16] proposed a method for learning a small pool of shared hypotheses in the context where many related tasks exist with few examples. This way, each task is then mapped to a single hypothesis in the learned pool (associating each with other related tasks). Thus, avoiding a possible inherent error that may occur in learning all the tasks together using a single hypothesis.

### 4.4 Decomposition

The decomposition method divides the weight matrix $\boldsymbol{W}$ into two or more component matrices (E.g., $\boldsymbol{W = D + B}$)), each of which can be penalized independently. As such, there are two main variations of which this method exists; the Dirty and Multilevel methods. While the dirty method decomposes the weight matrix into exactly two-component matrices, as shown in Fig. 5, the multilevel method decomposes the weight matrix into two or more component matrices. This way, each component matrix can then capture the various aspects of the task relationship. Hence, many MTL studies such as [37–39,93,96], utilized this technique. Illustratively, the least square convex optimization problem proposed in [38] is given as

$$\min_{\boldsymbol{S},\boldsymbol{B}} \frac{1}{2n} \sum_{k=1}^{r} \parallel y_k - X_k\left(S_k + B_k\right) \parallel_2^2 + \lambda_s \parallel S\parallel_{1,1} + \lambda_B \parallel \boldsymbol{B}\parallel_{1,\infty}, \tag{15}$$

where matrix $\boldsymbol{\Theta} \in R^{p \times r} = \boldsymbol{B} + \boldsymbol{S}$ based on the assumption that a certain number of rows in $\boldsymbol{\Theta}$ matrix will contain large non-zero entries, which correspond to the feature shared across

various tasks. Accordingly, some rows in $\boldsymbol{\Theta}$ matrix will also contain all-zero entries, which correspond to irrelevant features not needed by any task, while some rows will have elementwise sparseness corresponding to those features that are only relevant to some tasks but not all. Thus, matrices $\boldsymbol{B}$ and $\boldsymbol{S}$ capture a different aspect of relationship such that $\boldsymbol{S}$ captures elementwise sparsity whereas $\boldsymbol{B}$ captures row-wise sparsity with different regularization on both. Jalali et al. [37] then extended [38] by proposing a new forward–backward greedy procedure for the dirty model. The suggested technique identifies the best single variable and best row variable in each forward step that gives the largest incremental drop in the loss function. In contrast, it looks for the variable whose removal leads to the smallest incremental loss function rise in each backward step. Besides, a new adaptive method for multiple sparse linear regression was presented by Jalali et al. [39]. This approach was conceived by examining the multiple sparse linear regression problem, which entails recovering several related sparse vectors simultaneously. Thus, when there is support and parameter overlap, the proposed method takes advantage of it but does not pay the penalty when there isn't.

### 4.5 Tasks Similarity Learning

In this context, the pairwise relationships between tasks are learned directly from the data through a common model. Take as an example, when relying on the formulation in Eq. (1), then, the approach proposed in [22] is as follows

$$\begin{aligned} &\min_{w_0, v_t} \sum_{t=1}^{T} \sum_{i=1}^{N_t} \xi_{ti} + \tfrac{\lambda_1}{T} \sum_{t=1}^{T} \parallel v_t \parallel^2 + \lambda_2 \parallel w_0 \parallel^2, \\ &s.t., \; y_{ti}(w_0 + v_t) \cdot x_{ti} \geq 1 - \xi_{ti}, \; \xi_{ti} \geq 0, \forall t, \forall i, \end{aligned} \tag{16}$$

where $w_t$ is used to denote $w_0 + v_t$, $v_t$ is $t$ task-specific weight vector, and $w_0$ is common model between different tasks. The regularization constraint is imposed on $w_0$ (which captures the similarity between tasks) while constraining how much each $w_t$ vary from one another (allowing each $w_t$ to be close to some mean function $w_0$) by simultaneously controlling $v_t$'s size. Essentially, $v_t$ is smaller when tasks are related but, when $w_0 \to 0$, Eq. (16) reduces to an independent task problem where $w_t = v_t$. To further improve learning accuracy, Ji and Sun [42] extended the idea of [22] for non-linear MTL with a different task-specific base kernel. Since most previous multitask multiclass learning approaches aimed at decomposing multitask multiclass problems into multiple multitask binary, they do not completely capture the inherent correlations between classes. Therefore, a method was presented which can learn the multitask multiclass problems directly and efficiently. It was achieved by using a quadratic objective function to cast these problems into a constrained optimization one. Meanwhile, to capture negative task correlation and identify outlier tasks, Zhang et al. [92] proposed a method, which captures task relationship through a prior task covariance matrix obtained via the trace of a square matrix regularizer on weight matrix $\boldsymbol{W}$ as follows

$$tr(\boldsymbol{W}\Omega^{-1}\boldsymbol{W}^T), \tag{17}$$

where $tr(.)$ is the trace of a square matrix regularizer and $\Omega$ denotes a positive semi definite (PSD) tasks covariance matrix. Therefore, Eq. (17) is the same as the matrix fractional function given as:

$$\sum_{t} \boldsymbol{W}(t,:)\Omega^{-1}\boldsymbol{W}(t,:)^T, \tag{18}$$

where $\boldsymbol{W}(t,:)\Omega^{-1}$ denotes the $t$-th row of $\boldsymbol{W}$ matrix. Then by obtaining the Hessian matrix of $\boldsymbol{W}(t,:)\Omega^{-1}\boldsymbol{W}(t,:)^T$, Eq. (18) can be proved to be jointly convex w.r.t. $\boldsymbol{W}$, $\Omega$. Therefore,

**Table 1** A brief performance comparison of clustering, decomposition and tasks coupling methods using office-Caltech and MHC-I datasets with respect to classification error evaluation metric

| Dataset | Clustering | Decomposition | Tasks Similarity Learning |
|---|---|---|---|
| Office-Caltech | – | [38] (0.2030) | [90] (0.0690)/[23] (0.0450) |
| MHC-I | [36] (0.1890)/[44] (0.2050) | – | [90] (0.1870) |

Murugesan and Carbonell [62] extended the single kernel-based approach in [92] with task-specific multiple base kernels and proposed a method named Multitask Multiple Kernel Relationship Learning (MK-MTRL). MK-MTRL's main idea is to automatically assume task relationships in the RKHS space, similar to the one proposed in [32]. However, different from the work of [32], MK-MTRL formulation allows for incorporating prior knowledge to aid the simultaneous learning of several related tasks. Besides, Ruiz et al. [71] proposed a convex approach which can capture task relationship such that a convex penalty is imposed on both the task-specific weight $\| v_t \|^2$ and the common part $\| u \|^2$.

Also, Williams et al. [83] employed Gaussian processes to learn a task-similarity matrix with a block-diagonal structure that captures inter-task correlations by assuming tasks are ordered with regard to clusters. Consequently, a kernel-based method for automatically revealing structural inter-task relationships, which extend the low-rank output kernels strategy initially introduced in [21] to a multi-task environment, was proposed in the work of [20]. This approach uses a properly weighted loss, allowing several datasets with different input sampling patterns to be used. In another way, some efforts were made in [23,45] to capture the similarity between tasks using the Graph Laplacian strategy. Thus, guaranteeing that all tasks in the same cluster will have identical model parameters. Meanwhile, other efforts, such as [14,30,78] combined the ideas of numerous SSMTL techniques to increase generalization performance across all tasks.

Therefore, Table 1 gives brief performance comparison of Clustering, Decomposition and Tasks Coupling methods using Office-Caltech [38] and MHC-I [36] datasets.

## 5 Non-shallow Approach to SMTL

Before now, we focused on the Supervised Shallow approach to MTL, in which features are handcrafted according to the target problem. However, in a supervised deep learning paradigm, the best feature representation can be derived from the data directly using deep learning algorithms such as Convolutional Neural Network.

Therefore, Ruder [70] classified the deep efforts in MTL into hard and soft parameter sharing of hidden layers. The hard parameter method shown in Fig. 6 shares the hidden layers across several related tasks while keeping the tasks specific output layers. In contrast, the soft parameter-based method assigns to each task a specific model with its parameter. As a result, one can liken the soft margin approach to the shallow-based approach but, to capture relationships across multiple related tasks, the soft margin-based method obtains the distance between parameters of the different but related tasks, which are then regularized to encourage similarity.

MTL studies based on deep approach includes [24,61,69,79,84] with application area such as computer vision [24,69], speech synthesis [84] and bioinformatics-neuroanatomy [61,79]. All the same, [70,77] gave an extensive overview of deep MTL methods. And Fig. 7 shows

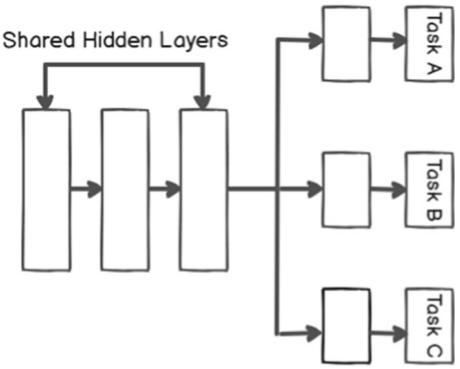


**Fig. 6** Hard parameter-based sharing of the hidden layers

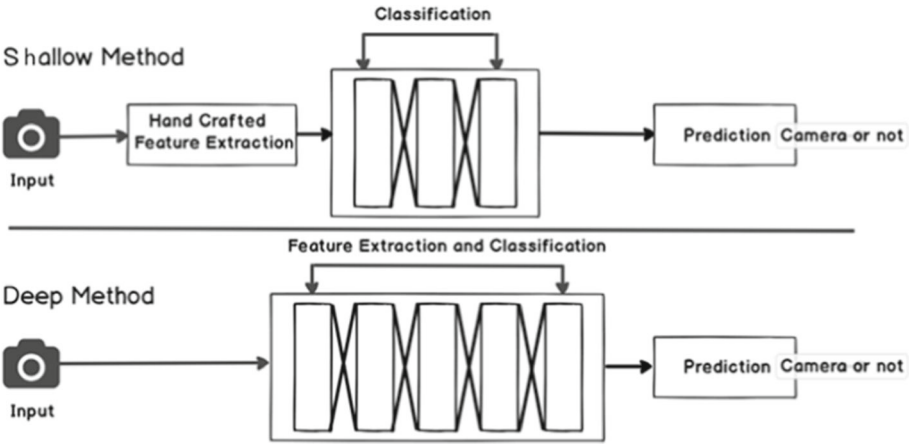


**Fig. 7** Shallow versus deep learning pipeline

a graphical comparison of Shallow vs. Deep Learning methods, while Table 2 gives a brief comparative analysis on both.

## 6 Challenges and Future Research Direction of SMTL

MTL aims to improve generalization performance by leveraging common information shared between related tasks. This way, the lost function is minimized on all similar tasks to obtain a unified model that generalizes to new tasks. At present, many studies have shown that MTL can provide robust improvement to single-task learning. Nonetheless, the generalization performance can degrade if a new task is unrelated or is an outlier to the model tasks. Besides, many existing MTL methods cannot guarantee that a trained unified MTL model will outperform the single-task model in all tasks. This is because an outlier task(s) can contribute negatively to learning the common information between related tasks. Although Zhang and Yang [88] had suggested an approach to tackle the first issue, it is not realistic in most real-world scenarios. For instance, while the suggested technique of detecting when a new task is not well-matched with the trained MTL model may be feasible, training another tasks model that matches the outlier task(s) will then present a new challenge. To address

**Table 2** A brief comparative analysis of shallow and deep learning approaches

| SN | Attributes | Shallow models | Deep models |
|---|---|---|---|
| 1 | Interpretability | It is much easy to predict a model's output even before the training begins | They usually lack a certain level of interpretability. Thus, it is hard to predict the model's outcome |
| 2 | Computational complexity | Complexity grows mainly as the data size increases | Complexity depends on both data and network sizes |
| 3 | Feature learning approach | Uses handcrafted features | Optimal features are learned directly from the data |
| 4 | Performance | Shallow models can maintain good performance even with a small amount of data | Deep models require a huge amount of data to obtain good performance |
| 5 | Flexibility | They may require complex extensions to solve newer problems | They can easily be adapted to new problems with limited changes |

these concerns holistically, there is a need to explore the combination of MTL and ensemble learning to learn common information shared between related tasks. This approach will improve generalization performance and further reduce the complexity of training a strong specific task model. Besides, task embeddings for MTL will be a fascinating area of research in the future. In this instance, tasks consistency can be addressed in order to preserve the geometric structure and information in each task to the greatest extent possible to help in learning a robust model that generalizes to newer tasks.

## 7 Conclusion

Most research work done on MTL focused on supervised learning, with several experimental results, which show that MTL is effective. Nevertheless, MTL based on unsupervised and reenforcement learning has recently gained more attention. Besides, few attempts exist to extend MTL to the semi-supervised learning paradigm such that MTL can benefit from incomplete data. In this paper, a review of existing supervised shallow-based MTL methods is made explicit, with specific attempts to present these methods without sophisticated mathematical deductions. Moreover, efforts were made to explain the concept of MTL with basic examples by avoiding ambiguity for readers.

## References

1. Abhadiomhen SE, Wang Z, Shen X (2021) Coupled low rank representation and subspace clustering. Appl Intell. https://doi.org/10.1007/s10489-021-02409-z
2. Abhadiomhen SE, Wang Z, Shen X, Fan J (2021) Multiview common subspace clustering via coupled low rank representation. ACM Trans Intell Syst Technol (TIST) 12(4):1–25
3. Ando RK, Zhang T (2005) A framework for learning predictive structures from multiple tasks and unlabeled data. J Mach Learn Res 6(11):1817–1853
4. Argyriou A, Evgeniou T, Pontil M (2007) Multi-task feature learning. In: Advances in neural information processing systems. pp 41–48
5. Argyriou A, Evgeniou T, Pontil M (2008) Convex multi-task feature learning. Mach Learn 73(3):243–272

6. Aydin E, Yüksel SE (2018) Transfer and multitask learning method for buried wire detection via gpr. In: 2018 26th Signal processing and communications applications conference (SIU). IEEE, pp 1–4
7. Bakker B, Heskes T (2003) Task clustering and gating for Bayesian multitask learning. J Mach Learn Res 4(May):83–99
8. Baxter J (2000) A model of inductive bias learning. J Artif Intell Res 12:149–198
9. Ben-David S, Borbely RS (2008) A notion of task relatedness yielding provable multiple-task learning guarantees. Mach Learn 73(3):273–287
10. Cai F, Cherkassky V (2009) Svm+ regression and multi-task learning. In: 2009 International joint conference on neural networks. IEEE, pp 418–424
11. Caruana R (1997) Multitask learning. Mach Learn 28(1):41–75
12. Chapelle O, Shivaswamy P, Vadrevu S, Weinberger K, Zhang Y, Tseng B (2011) Boosted multi-task learning. Mach Learn 85(1–2):149–173
13. Chen J, Tang L, Liu J, Ye J (2009) A convex formulation for learning shared structures from multiple tasks. In: Proceedings of the 26th annual international conference on machine learning. pp 137–144
14. Chen J, Liu J, Ye J (2012) Learning incoherent sparse and low-rank patterns from multiple tasks. ACM Trans Knowl Discov Data (TKDD) 5(4):1–31
15. Chen X, Pan W, Kwok JT, Carbonell JG (2009) Accelerated gradient method for multi-task sparse learning problem. In: 2009 Ninth IEEE international conference on data mining. IEEE, pp 746–751
16. Crammer K, Mansour Y (2012) Learning multiple tasks using shared hypotheses. Adv Neural Inf Process Syst 25:1475–1483
17. Crammer K, Singer Y (2001) On the algorithmic implementation of multiclass kernel-based vector machines. J Mach Learn Res 2(12):265–292
18. Dai Y, Zhang J, Yuan S, Xu Z (2019) A two-stage multi-task learning-based method for selective unsupervised domain adaptation. In: 2019 International conference on data mining workshops (ICDMW). IEEE, pp 863–868
19. Dereli O, Oğuz C, Gönen M (2019) A multitask multiple kernel learning algorithm for survival analysis with application to cancer biology. In: International conference on machine learning. pp 1576–1585
20. Dinuzzo F (2013) Learning output kernels for multi-task problems. Neurocomputing 118:119–126
21. Dinuzzo F, Fukumizu K (2011) Learning low-rank output kernels. In: Asian conference on machine learning, PMLR. pp 181–196
22. Evgeniou T, Pontil M (2004) Regularized multi-task learning. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. pp 109–117
23. Evgeniou T, Micchelli CA, Pontil M (2005) Learning multiple tasks with kernel methods. J Mach Learn Res 6(4):615–637
24. Fang Y, Ma Z, Zhang Z, Zhang XY, Bai X et al (2017) Dynamic multi-task learning with convolutional neural network. In: IJCAI. pp 1668–1674
25. Gong P, Ye J, Zhang C (2013) Multi-stage multi-task feature learning. J Mach Learn Res 14(1):2979–3010
26. Gong P, Zhou J, Fan W, Ye J (2014) 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
27. Goussies NA, Ubalde S, Mejail M (2014) Transfer learning decision forests for gesture recognition. J Mach Learn Res 15(1):3667–3690
28. Gu Q, Li Z, Han J (2011) Learning a kernel for multi-task clustering. In: AAAI. pp 368–373
29. Han L, Zhang Y (2016) Multi-stage multi-task learning with reduced rank. In: AAAI. pp 1638–1644
30. He J, Lawrence R (2011) A graphbased framework for multi-task multi-view learning. In: ICML. pp 25–32
31. He Z, Li J, Liu L (2017) Hyperspectral classification based on kernel low-rank multitask learning. In: 2017 IEEE international geoscience and remote sensing symposium (IGARSS). IEEE, pp 3206–3209
32. He Z, Li J, Liu K, Liu L, Tao H (2018) Kernel low-rank multitask learning in variational mode decomposition domain for multi-/hyperspectral classification. IEEE Trans Geosci Remote Sens 56(7):4193–4208
33. Huang J, Qin F, Zheng X, Cheng Z, Yuan Z, Zhang W (2018) Learning label-specific features for multi-label classification with missing labels. In: 2018 IEEE fourth international conference on multimedia big data (BigMM). IEEE, pp 1–5
34. Huang M, Zhuang F, Zhang X, Ao X, Niu Z, Zhang ML, He Q (2019) Supervised representation learning for multi-label classification. Mach Learn 108(5):747–763
35. Jacob L, Bach F, Vert JP (2008) Clustered multi-task learning: a convex formulation. arXiv preprint arXiv:0809.2085
36. Jacob L, Vert J, Bach FR (2009) Clustered multi-task learning: a convex formulation. In: Advances in neural information processing systems. pp 745–752
37. Jalali A, Sanghavi S (2012) Greedy dirty models: a new algorithm for multiple sparse regression. In: 2012 IEEE statistical signal processing workshop (SSP). IEEE, pp 416–419

38. Jalali A, Sanghavi S, Ruan C, Ravikumar PK (2010) A dirty model for multi-task learning. In: Advances in neural information processing systems. pp 964–972
39. Jalali A, Ravikumar P, Sanghavi S (2013) A dirty model for multiple sparse regression. IEEE Trans Inf Theory 59(12):7947–7968
40. Jawanpuria P, Nath JS (2011) Multi-task multiple kernel learning. In: Proceedings of the 2011 SIAM international conference on data mining. SIAM, pp 828–838
41. Jawanpuria P, Jagarlapudi SN, Ramakrishnan G (2011) Efficient rule ensemble learning using hierarchical kernels. In: Proceedings of the 28th international conference on machine learning (ICML-11). pp 161–168
42. Ji Y, Sun S (2013) Multitask multiclass support vector machines: model and experiments. Pattern Recognit 46(3):914–924
43. Jiangmei Z, Binfeng Y, Haibo J, Wang K (2017) Multi-task feature learning by using trace norm regularization. Open Phys 15(1):674–681
44. Kang Z, Grauman K, Sha F (2011) Learning with whom to share in multi-task feature learning. In: ICML, vol 2. p 4
45. Kato T, Kashima H, Sugiyama M, Asai K (2008) Multi-task learning via conic programming. In: Advances in neural information processing systems. pp 737–744
46. Kendall A, Gal Y, Cipolla R (2018) Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 7482–7491
47. Kumar A, Daumé H (2012) Learning task grouping and overlap in multi-task learning. In: Proceedings of the 29th international conference on machine learning. Omnipress, Madison, WI, USA, ICML'12. pp 1723–1730
48. Lei Y, Binder A, Dogan Ü, Kloft M (2015) Theory and algorithms for the localized setting of learning kernels. In: Feature extraction: modern questions and challenges. pp 173–195
49. Li Y, Wang J, Ye J, Reddy CK (2016) A multi-task learning formulation for survival analysis. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. pp 1715–1724
50. Liu G, Lin Z, Yan S, Sun J, Yu Y, Ma Y (2013) Robust recovery of subspace structures by low-rank representation. IEEE Trans Pattern Anal Mach Intell 35(1):171–184
51. Liu J, Ji S, Ye J (2009) Multi-task feature learning via efficient l2, 1-norm minimization. In: Proceedings of the twenty th conference on uncertainty in artificial intelligence. AUAI Press, pp 339–348
52. Liu T, Tao D, Song M, Maybank SJ (2017) Algorithm-dependent generalization bounds for multi-task learning. IEEE Trans Pattern Anal Mach Intell 39(2):227–241
53. Liu X, Zhu X, Li M, Wang L, Zhu E, Liu T, Kloft M, Shen D, Yin J, Gao W (2019) Multiple kernel $k$ k-means with incomplete kernels. IEEE Trans Pattern Anal Mach Intell 42(5):1191–1204
54. Lopez-Martinez D (2017) Regularization approaches for support vector machines with applications to biomedical data. arXiv preprint arXiv:1710.10600
55. Lounici K, Pontil M, Tsybakov AB, Van De Geer S (2009) Taking advantage of sparsity in multi-task learning. arXiv preprint arXiv:0903.1468
56. Lozano AC, Swirszcz G (2012) Multi-level lasso for sparse multi-task regression. In: Proceedings of the 29th international conference on machine learning. pp 595–602
57. Maurer A, Pontil M (2010) $k$-dimensional coding schemes in Hilbert spaces. IEEE Trans Inf Theory 56(11):5839–5846
58. Maurer A, Pontil M, Romera-Paredes B (2016) The benefit of multitask representation learning. J Mach Learn Res 17(1):2853–2884
59. McDonald AM, Pontil M, Stamos D (2014) Spectral k-support norm regularization. In: Advances in neural information processing systems. pp 3644–3652
60. Mei B, Xu Y (2019) Multi-task least squares twin support vector machine for classification. Neurocomputing 338:26–33
61. Moeskops P, Wolterink JM, van der Velden BH, Gilhuijs KG, Leiner T, Viergever MA, Išgum I (2016) Deep learning for multi-task medical image segmentation in multiple modalities. In: International conference on medical image computing and computer-assisted intervention. Springer, pp 478–486
62. Murugesan K, Carbonell J (2017) Multi-task multiple kernel relationship learning. In: Proceedings of the 2017 SIAM international conference on data mining. SIAM, pp 687–695
63. Naik SM, Jagannath RPK (2017) Accurate validation of GCV-based regularization parameter for extreme learning machine. In: 2017 International conference on advances in computing, communications and informatics (ICACCI). IEEE, pp 1727–1731
64. Obozinski G, Taskar B, Jordan M (2006) Multi-task feature selection. Statistics Department, UC Berkeley, Tech Rep 2(2.2):2

65. Oh J, Singh S, Lee H, Kohli P (2017) Zero-shot task generalization with multi-task deep reinforcement learning. In: Proceedings of the 34th international conference on machine learning, vol 70. pp 2661–2670
66. Parameswaran S, Weinberger KQ (2010) Large margin multi-task metric learning. In: Advances in neural information processing systems. pp 1867–1875
67. Pong TK, Tseng P, Ji S, Ye J (2010) Trace norm regularization: reformulations, algorithms, and multi-task learning. SIAM J Optim 20(6):3465–3489
68. Rakotomamonjy A, Bach FR, Canu S, Grandvalet Y (2008) Simplemkl. J Mach Learn Res 9(11):2491–2521
69. Ranjan R, Patel VM, Chellappa R (2017) Hyperface: a deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. IEEE Trans Pattern Anal Mach Intell 41(1):121–135
70. Ruder S (2017) An overview of multi-task learning in deep neural networks. arXiv preprint arXiv:1706.05098
71. Ruiz C, Alaíz CM, Dorronsoro JR (2019) A convex formulation of SVM-based multi-task learning. In: International conference on hybrid artificial intelligence systems. Springer, pp 404–415
72. Shrivastava A, Patel VM, Chellappa R (2014) Multiple kernel learning for sparse representation-based classification. IEEE Trans Image Process 23(7):3013–3024
73. Standley T, Zamir A, Chen D, Guibas L, Malik J, Savarese S (2020) Which tasks should be learned together in multi-task learning? In: International conference on machine learning, PMLR. pp 9120–9132
74. Sun T, Jiao L, Liu F, Wang S, Feng J (2013) Selective multiple kernel learning for classification with ensemble strategy. Pattern Recognit 46(11):3081–3090
75. Suzuki T, Tomioka R (2011) Spicymkl: a fast algorithm for multiple kernel learning with thousands of kernels. Mach Learn 85(1–2):77–108
76. Thrun S, O'Sullivan J (1996) Discovering structure in multiple learning tasks: the TC algorithm. ICML 96:489–497
77. Thung KH, Wee CY (2018) A brief review on multi-task learning. Multimedia Tools Appl 77(22):29705–29725
78. Tian X, Li Y, Liu T, Wang X, Tao D (2019) Eigenfunction-based multitask learning in a reproducing kernel Hilbert space. IEEE Trans Neural Netw Learn Syst 30(6):1818–1830
79. Wachinger C, Reuter M, Klein T (2018) Deepnat: deep convolutional neural network for segmenting neuroanatomy. NeuroImage 170:434–445
80. Wang L, Zhu J, Zou H (2006) The doubly regularized support vector machine. Stat Sin 16:589–615
81. Wang X, Bi J, Yu S, Sun J, Song M (2016) Multiplicative multitask feature learning. J Mach Learn Res 17(1):2820–2852
82. Widmer C, Toussaint NC, Altun Y, Rätsch G (2010) Inferring latent task structure for multitask learning by multiple kernel learning. BMC Bioinform 11(8):1–8
83. Williams C, Bonilla EV, Chai KM (2007) Multi-task gaussian process prediction. In: Advances in neural information processing systems. pp 153–160
84. Wu Z, Valentini-Botinhao C, Watts O, King S (2015) Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis. In: 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 4460–4464
85. Xiao Y, Chang Z, Liu B (2020) An efficient active learning method for multi-task learning. Knowl Based Syst 190:105137
86. Yu X, Zhou Z, Gao Q, Li D, Ríha K (2018) Infrared image segmentation using growing immune field and clone threshold. Infrared Phys Technol 88:184–193
87. Zhang T et al (2013) Multi-stage convex relaxation for feature selection. Bernoulli 19(5B):2277–2293
88. Zhang Y, Yang Q (2017) A survey on multi-task learning. arXiv preprint arXiv:1707.08114
89. Zhang Y, Yang Q (2018) An overview of multi-task learning. Natl Sci Rev 5(1):30–43
90. Zhang Y, Yeung DY (2010) A convex formulation for learning task relationships in multi-task learning. In: Proceedings of the 26th conference on uncertainty in artificial intelligence, vol 7. pp 33–42
91. Zhang Y, Yeung DY (2010) Multi-task learning using generalized t process. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. pp 964–971
92. Zhang Y, Yeung DY (2014) A regularization approach to learning task relationships in multitask learning. ACM Trans Knowl Discov Data (TKDD) 8(3):1–31
93. Zhong W, Kwok J (2012) Convex multitask learning with flexible task clusters. In: Proceedings of the 29th international conference on machine learning
94. Zhou Q, Chen Y, Pan SJ (2020) Communication-efficient distributed multi-task learning with matrix sparsity regularization. Mach Learn 109:1–33
95. Zhou Y, Jin R, Hoi SCH (2010) Exclusive lasso for multi-task feature selection. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. pp 988–995

96. Zweig A, Weinshall D (2013) Hierarchical regularization cascade for joint learning. In: International conference on machine learning. pp 37–45