REVIEW

National Science Review
5: 30–43, 2018
doi: 10.1093/nsr/nwx105
Advance access publication 1 September 2017

COMPUTER SCIENCE

Special Topic: Machine Learning

An overview of multi-task learning

Yu Zhang* and Qiang Yang*

ABSTRACT

As a promising area in machine learning, multi-task learning (MTL) aims to improve the performance of multiple related learning tasks by leveraging useful information among them. In this paper, we give an overview of MTL by first giving a definition of MTL. Then several different settings of MTL are introduced, including multi-task supervised learning, multi-task unsupervised learning, multi-task semi-supervised learning, multi-task active learning, multi-task reinforcement learning, multi-task online learning and multi-task multi-view learning. For each setting, representative MTL models are presented. In order to speed up the learning process, parallel and distributed MTL models are introduced. Many areas, including computer vision, bioinformatics, health informatics, speech, natural language processing, web applications and ubiquitous computing, use MTL to improve the performance of the applications involved and some representative works are reviewed. Finally, recent theoretical analyses for MTL are presented.

Keywords: multi-task learning

INTRODUCTION

Machine learning, which exploits useful information in historical data and utilizes the information to help analyze future data, usually needs a large amount of labeled data for training a good learner. One typical learner in machine learning is deep-learning models, which are neural networks with many hidden layers and also many parameters; these models usually need millions of data instances to learn accurate parameters. However, some applications such as medical image analysis cannot satisfy this requirement since it needs more manual labor to label data instances. In these cases, multi-task learning (MTL) [1] is a good recipe by exploiting useful information from other related learning tasks to help alleviate this data sparsity problem.

As a promising area in machine learning, MTL aims to leverage useful information contained in multiple learning tasks to help learn a more accurate learner for each task. Based on an assumption that all the tasks, or at least a subset of them, are related, jointly learning multiple tasks is empirically and theoretically found to lead to better performance than learning them independently. Based on the nature of the tasks, MTL can be classified into several settings, including multi-task supervised learning, multi-task unsupervised learning, multi-task semi-supervised learning, multi-task active learning, multi-task reinforcement learning, and multi-task online learning. In multi-task supervised learning, each task, which can be a classification or regression problem, is to predict labels for unseen data instances given a training dataset consisting of training data instances and their labels. In multi-task unsupervised learning, each task, which can be a clustering problem, aims to identify useful patterns contained in a training dataset consisting of data instances only. In multi-task semi-supervised learning, each task is similar to that in multi-task supervised learning with the difference that the training set includes not only labeled data but also unlabeled ones. In multi-task active learning, each task exploits unlabeled data to help learn from labeled data similar to multi-task semi-supervised learning but in a different way by selecting unlabeled data instances to actively query their labels. In multi-task reinforcement learning, each task aims to choose actions to maximize the cumulative reward. In multi-task online learning, each task handles sequential data. In multi-task multi-view learning, each task handles

Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, China

*Corresponding authors. E-mails: yuzhangcse@cse.ust.hk, qyang@cse.ust.hk

Received 27 June 2017; **Revised** 22 July 2017; **Accepted** 8 August 2017

multi-view data in which there are multiple sets of features to describe each data instance.

MTL can be viewed as one way for machines to mimic human learning activities since people often transfer knowledge from one task to another and vice versa when these tasks are related. One example from our own experience is that the skills for playing squash and tennis can help improve each other. Similar to human learning, it is useful to learn multiple learning tasks simultaneously since the knowledge in a task can be utilized by other related tasks.

MTL is related to other areas in machine learning, including transfer learning [2], multi-label learning [3] and multi-output regression, but exhibits different characteristics. For example, similar to MTL, transfer learning also aims to transfer knowledge from one task to another but the difference lies in that transfer learning hopes to use one or more tasks to help a target task while MTL uses multiple tasks to help each other. When different tasks in multi-task supervised learning share the training data, it becomes multi-label learning or multi-output regression. In this sense, MTL can be viewed as a generalization of multi-label learning and multi-output regression.

In this paper, we give an overview of MTL. We first briefly introduce MTL by giving its definition. After that, based on the nature of each learning task, we discuss different settings of MTL, including multi-task supervised learning, multi-task unsupervised learning, multi-task semi-supervised learning, multi-task active learning, multi-task reinforcement learning, multi-task online learning and multi-task multi-view learning. For each setting of MTL, representative MTL models are presented. When the number of tasks is large or data in different tasks are located in different machines, parallel and distributed MTL models become necessary and several models are introduced. As a promising learning paradigm, MTL has been applied to several areas, including computer vision, bioinformatics, health informatics, speech, natural language processing, web applications and ubiquitous computing, and several representative applications in each area are presented. Moreover, theoretical analyses for MTL, which can give us a deep understanding of MTL, are reviewed.

The remainder of this paper is organized as follows. The section entitled 'Multi-task learning' introduces the definition of MTL. From the section entitled 'Multi-task supervised learning' to that entitled 'Multi-task multi-view learning', we give an overview of different settings in MTL, including multi-task supervised learning, multi-task unsupervised learning, multi-task semi-supervised learning, multi-task active learning, multi-task reinforcement learning, multi-task online learning and multi-task multi-view learning. The section entitled 'Parallel and distributed MTL' discusses parallel and distributed MTL models. The section entitled 'Applications of multi-task learning' shows how MTL can help other areas and that entitled 'Theoretical analysis' focuses on theoretical analyses of MTL. Finally, the section entitled 'Conclusions' concludes the whole paper.[1]

MULTI-TASK LEARNING

To start with, we give a definition of MTL.

Definition 1. (**Multi-task learning**) Given m learning tasks $\{\mathcal{T}_i\}_{i=1}^m$ where all the tasks or a subset of them are related but not identical, *multi-task learning* aims to help improve the learning of a model for $\mathcal{T}_i$ by using the knowledge contained in the m tasks.

Based on this definition, we can see that there are two elementary factors for MTL.

The first factor is the task relatedness. The task relatedness is based on the understanding of how different tasks are related, which will be encoded into the design of MTL models, as we will see later.

The second factor is the definition of task. In machine learning, learning tasks mainly include supervised tasks such as classification and regression tasks, unsupervised tasks such as clustering tasks, semi-supervised tasks, active learning tasks, reinforcement learning tasks, online learning tasks and multi-view learning tasks. Hence different learning tasks lead to different settings in MTL, which is what the following sections focus on. In the following sections, we will review representative MTL models in different MTL settings.

MULTI-TASK SUPERVISED LEARNING

The multi-task supervised learning (MTSL) setting means that each task in MTL is a supervised learning task, which models the functional mapping from data instances to labels. Mathematically, suppose there are m supervised learning tasks $\mathcal{T}_i$ for $i = 1, \ldots, m$ and each supervised task is associated with a training dataset $\mathcal{D}_i = \{(\mathbf{x}_j^i, y_j^i)\}_{j=1}^{n_i}$, where each data instance $\mathbf{x}_j^i$ lies in a d-dimensional space and y_j^i is the label for $\mathbf{x}_j^i$. So, for the ith task $\mathcal{T}_i$, there are n_i pairs of data instances and labels. When y_j^i is in a continuous space or equivalently a real scalar, the corresponding task is a regression task and if y_j^i is discrete, i.e. $y_j^i \in \{-1, 1\}$, the corresponding task is a classification task.

[1] For a more technical or complete survey on MTL, please refer to [4].

MTSL aims to learn m functions $\{f_i(\mathbf{x})\}_{i=1}^m$ for the m tasks from the training set such that $f_i(\mathbf{x}_j^i)$ is a good approximation of y_j^i for all the i and j. After learning the m functions, MTSL uses $f_i(\cdot)$ to predict labels of unseen data instances from the ith task.

As discussed before, the understanding of task relatedness affects the design of MTSL models. Specifically, existing MTSL models reflect the task relatedness in three aspects: feature, parameter and instance, leading to three categories of MTSL models including feature-based, parameter-based, and instance-based MTSL models. Specifically, feature-based MTSL models assume that different tasks share identical or similar feature representations, which can be a subset or a transformation of the original features. Parameter-based MTSL models aim to encode the task relatedness into the learning model via the regularization or prior on model parameters. Instance-based MTSL models propose to use data instances from all the tasks to construct a learner for each task via instance weighting. In the following, we will review representative models in the three categories.

Feature-based MTSL

In this category, all MTL models assume that different tasks share a feature representation, which is induced by the original feature representation. Based on how the shared feature representation appears, we further categorize multi-task models into three approaches, including the feature transformation approach, the feature selection approach and the deep-learning approach. The feature transformation approach learns the shared feature representation as a linear or nonlinear transformation of the original features. The feature selection approach assumes that the shared feature representation is a subset of the original features. The deep-learning approach applies deep neural networks to learn the shared feature representation, which is encoded in the hidden layers, for multiple tasks.

Feature transformation approach

In this approach, the shared feature representation is a linear or nonlinear transformation of the original feature representation. A representative model is the multi-layer feedforward neural network [1] and an example of a multi-layer feedforward neural network is shown in Fig. 1. In this example, the multi-layer feedforward neural network consists of an input layer, a hidden layer, and an output layer. The input layer has d units to receive data instances from the m tasks as inputs with one unit for a feature. The hidden layer contains multiple nonlinear activation units and receives the transformed output of the input layer as the input where the transformation depends on the weights connecting the input and hidden layers. As a transformation of the original features, the output of the hidden layer is the feature representation shared by all the tasks. The output of the hidden layer is first transformed based on the weights connecting the hidden and output layers, and then fed into the output layer, which has m units, each of which corresponds to a task.

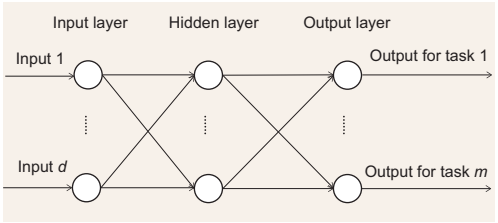

Figure 1. A multi-task feedforward neural network with one input layer, hidden layer and output layer.

Unlike multi-layer feedforward neural networks, which are based on neural networks, the multi-task feature learning (MTFL) method [5,6] and the multi-task sparse coding (MTSC) method [7] are formulated under the regularization framework by first transforming data instances as $\hat{\mathbf{x}}_j^i = \mathbf{U}^T\mathbf{x}_j^i$ and then learning a linear function as $f_i(\mathbf{x}_j^i) = (\mathbf{a}^i)^T\hat{\mathbf{x}}_j^i + b_i$. Based on this formulation, we can see that these two methods aim to learn a linear transformation $\mathbf{U}$ instead of the nonlinear transformation in multi-layer feedforward neural networks. Moreover, for the MTFL and MTSC methods, there exist several differences. For example, in the MTFL method, $\mathbf{U}$ is supposed to be orthogonal and the parameter matrix $\mathbf{A} = (\mathbf{a}^1, \ldots, \mathbf{a}^m)$ is row-sparse via the $\ell_{2,1}$ regularization, while in the MTSC method, $\mathbf{U}$ is overcomplete, implying that the number of columns in $\mathbf{U}$ is much larger than the number of rows, and $\mathbf{A}$ is sparse via the ℓ_1 regularization.

Feature selection approach

The feature selection approach aims to select a subset of original features as the shared feature representation for different tasks. There are two ways to do the multi-task feature selection. The first way is based on the regularization on $\mathbf{W} = (\mathbf{w}^1, \ldots, \mathbf{w}^m)$, where $f_i(\mathbf{x}) = (\mathbf{w}^i)^T\mathbf{x} + b_i$ defines the linear learning function for $\mathcal{T}_i$, and another one is based on sparse probabilistic priors on $\mathbf{W}$. In the following, we will give details of these two ways.

Among all the regularized methods for multi-task feature selection, the most widely used technique is $\ell_{p,q}$ regularization to minimize $\|\mathbf{W}\|_{p,q}$, the $\ell_{p,q}$ norm of $\mathbf{W}$, plus the training loss on the training

set, where $\mathbf{w}_j$ denotes the jth row of $\mathbf{W}$, $\|\cdot\|_q$ denotes the ℓ_q norm of a vector, and $\|\mathbf{W}\|_{p,q}$ equals $\left\|\left(\|\mathbf{w}_1\|_p, \ldots, \|\mathbf{w}_d\|_p\right)\right\|_q$. The effect of the $\ell_{p,q}$ regularization is to make $\mathbf{W}$ row-sparse and hence some unimportant features for all the tasks can be filtered out. Concrete instances of the $\ell_{p,q}$ regularization include the $\ell_{2,1}$ regularization proposed in [8,9] and the $\ell_{\infty,1}$ regularization proposed in [10]. In order to obtain a smaller subset of useful features for multiple tasks, a capped-$\ell_{p,1}$ penalty, which is defined as $\sum_{i=1}^{d}\min(\|\mathbf{w}_i\|_p, \theta)$, is proposed in [11]. It is easy to see that when θ becomes large enough, this capped-$\ell_{p,1}$ penalty will degenerate to the $\ell_{p,1}$ regularization. Besides the $\ell_{p,q}$ regularization, there is another type of regularized method, which can select a feature for MTL. For example, in [12], a multi-level lasso is proposed by decomposing w_{ji}, the (j,i)th entry in $\mathbf{W}$, as $w_{ji} = \theta_j \hat{w}_{ji}$. It is easy to see that when θ_j equals 0, $\mathbf{w}_j$ becomes a zero row, implying that the jth feature is not useful for all the tasks, and hence θ_j is an indicator of the usefulness of the jth feature for all the tasks. Moreover, when $\hat{w}_{ji}$ becomes 0, w_{ji} will also become 0 and hence $\hat{w}_{ji}$ is an indicator of the usefulness of the jth feature for $\mathcal{T}_i$ only. By regularizing θ_j and $\hat{w}_{ji}$ via the ℓ_1 norm to enforce them to be sparse, the multi-level lasso can learn sparse features in two levels. This model is extended in [13,14] to more general settings.

For multi-task feature selection methods based on the $\ell_{p,1}$ regularization, a probabilistic interpretation is proposed in [15], which shows that the $\ell_{p,1}$ regularizer corresponds to a prior: $w_{ji} \sim \mathcal{GN}(0, \rho_j, p)$, where $\mathcal{GN}(\cdot,\cdot,\cdot)$ denotes the generalized normal distribution. Then this prior is extended in [15] to the matrix-variate generalized normal prior to learn relations among tasks and identify outlier tasks simultaneously. In [16,17], the horseshoe prior is utilized to select features for MTL. The difference between [16] and [17] is that in [16], the horseshoe prior is generalized to learn feature covariance, while in [17], the horseshoe prior is used as a basic prior and the whole model is to identify outlier tasks in a way different from [15].

Deep-learning approach

Similar to the multi-layer feedforward neural network model in the feature transformation approach, basic models in the deep-learning approach include advanced neural network models such as convolutional neural networks and recurrent neural networks. However, unlike the multi-layer feedforward neural network with a small number of hidden layers (e.g. 2 or 3), the deep-learning approach involves neural networks with tens of or even hundreds of hidden layers. Moreover, similar to the multi-layer feedforward neural network, most deep-learning models [18–22] in this category treat the output of one hidden layer as the shared feature representation. Unlike these deep models, the cross-stitch network proposed in [23] combines the hidden feature representations of two tasks to construct more powerful hidden feature representations. Specifically, given two deep neural networks A and B with the same network architecture for two tasks, where $x^A_{i,j}$ and $x^B_{i,j}$ denote the hidden features contained in the jth unit of the ith hidden layer for networks A and B, the cross-stitch operation on $x^A_{i,j}$ and $x^B_{i,j}$ can be defined as

$$\begin{pmatrix} \tilde{x}^A_{i,j} \\ \tilde{x}^B_{i,j} \end{pmatrix} = \begin{pmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{pmatrix} \begin{pmatrix} x^A_{i,j} \\ x^B_{i,j} \end{pmatrix},$$

where $\tilde{x}^A_{i,j}$ and $\tilde{x}^B_{i,j}$ denote new hidden features after the joint learning of the two tasks. Matrix $\boldsymbol{\alpha} = \begin{pmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{pmatrix}$ as well as the parameters in the two networks are learned from data via the back propagation method and hence this method is more flexible than directly sharing hidden layers.

Parameter-based MTSL

Parameter-based MTSL uses model parameters to relate the learning of different tasks. Based on how the model parameters of different tasks are related, we classify them into five approaches, including the low-rank approach, the task-clustering approach, the task-relation learning approach, the dirty approach and the multi-level approach. Specifically, since tasks are assumed to be related, the parameter matrix $\mathbf{W}$ is likely to be low-rank, which is the motivation for the low-rank approach. The task-clustering approach aims to divide tasks into several clusters and all the tasks in a cluster are assumed to share identical or similar model parameters. The task-relation learning approach directly learns the pairwise task relations from data. The dirty approach assumes the decomposition of the parameter matrix $\mathbf{W}$ into two component matrices, each of which is regularized by a type of the sparsity. As a generalization of the dirty approach, the multi-level approach decomposes the parameter matrix into more than 2 component matrices to model complex relations among all the tasks. In the following sections, we will discuss each approach in detail.

Low-rank approach

Similar tasks usually have similar model parameters, which makes $\mathbf{W}$ likely to be low-rank. In [24], the model parameters of the m tasks are assumed to share a low-rank subspace, leading to a

parametrization of $\mathbf{w}^i$ as $\mathbf{w}^i = \mathbf{u}^i + \mathbf{\Theta}^T\mathbf{v}^i$, where $\mathbf{\Theta} \in \mathbb{R}^{h\times d}$ is a low-rank subspace shared by all the tasks with $h < d$ and $\mathbf{u}^i$ is specific to task $\mathcal{T}_i$. With an assumption on $\mathbf{\Theta}$ that $\mathbf{\Theta}$ is orthonormal (i.e. $\mathbf{\Theta}\mathbf{\Theta}^T = \mathbf{I}$ where $\mathbf{I}$ denotes an identity matrix with an appropriate size) to remove the redundancy, $\mathbf{u}^i$, $\mathbf{v}^i$ and $\mathbf{\Theta}$ are learned by minimizing the training loss on all the tasks. This model is then generalized in [25] by adding a squared Frobenius regularization on $\mathbf{W}$ and this generalized model can be relaxed to have a convex objective function.

Based on the analysis in optimization, regularizing with the trace norm, which is defined as $\|\mathbf{W}\|_{S(1)} = \sum_{i=1}^{\min(m,d)} \mu_i(\mathbf{W})$, can make a matrix low-rank and hence trace-norm regularization is widely used in MTL with [26] as a representative work. Similar to what the capped-$\ell_{p,1}$ penalty did to the $\ell_{p,1}$ norm, a variant of the trace-norm regularization called the capped-trace regularizer is proposed in [27] and defined as $\sum_{i=1}^{\min(m,d)} \min(\mu_i(\mathbf{W}), \theta)$, where θ is a parameter defined by users. Based on θ, only small singular values of $\mathbf{W}$ will be penalized and hence it can lead to a matrix with a lower rank. When θ becomes large enough, the capped-trace regularizer will reduce to the trace norm.

Task-clustering approach

The task-clustering approach applies the idea of data-clustering methods to group tasks into several clusters, each of which has similar tasks in terms of model parameters.

The first task-clustering algorithm proposed in [28] decouples the task-clustering procedure and the model-learning procedure. Specifically, it first clusters tasks based on the model parameters learned separately under the single-task setting and then pools the training data of all the tasks in a task cluster to learn a more accurate learner for all the tasks in this task cluster. This two-stage method may be suboptimal since model parameters learned under the single-task setting may be inaccurate, making the task-clustering procedure not so good. So follow-up research aims to identify the task clusters and learn model parameters together.

A multi-task Bayesian neural network, whose structure is similar to that of the multi-layer neural network shown in Fig. 1, is proposed in [29] to cluster tasks based on the Gaussian mixture model in terms of model parameters (i.e. weights connecting the hidden and output layers). The Dirichlet process, which is widely used in Bayesian learning to do data clustering, is employed in [30] to do task clustering based on model parameters $\{\mathbf{w}^i\}$.

Unlike [29,30], which are Bayesian models, there are several regularized methods [31–35] to do task clustering. Inspired by the k-means clustering method, Jacob et al. [31] devise a regularizer, i.e. $\text{tr}(\mathbf{W}\mathbf{\Pi}\mathbf{\Sigma}^{-1}\mathbf{\Pi}\mathbf{W}^T)$, to identify task clusters by considering between-cluster and within-cluster variances, where $\text{tr}(\cdot)$ gives the trace of a square matrix, $\mathbf{\Pi}$ denotes an $m \times m$ centering matrix, $\mathbf{A} \preceq \mathbf{B}$ for two square matrices $\mathbf{A}, \mathbf{B}$ means that $\mathbf{B} - \mathbf{A}$ is positive semidefinite (PSD), and with three hyperparameters α, β, γ, $\mathbf{\Sigma}$ is required to satisfy $\alpha\mathbf{I} \preceq \mathbf{\Sigma} \preceq \beta\mathbf{I}$ and $\text{tr}(\mathbf{\Sigma}) = \gamma$. The MTFL method is extended in [32] to the case of multiple clusters, where each cluster applies the MTFL method, and in order to learn the cluster structure, a regularizer, i.e. $\sum_{i=1}^{r} \|\mathbf{W}\mathbf{Q}_i\|_{S(1)}^2$, is employed, where a 0/1 diagonal matrix $\mathbf{Q}_i$ satisfying $\sum_{i=1}^{r} \mathbf{Q}_i = \mathbf{I}$ can help identify the structure of the ith cluster. In order to automatically determine the number of clusters, a structurally sparse regularizer, $\sum_{j>i} \|\mathbf{w}^i - \mathbf{w}^j\|_2$, is proposed in [34] to enforce any pair of model parameters to be fused. After learning the parameter matrix $\mathbf{W}$, the cluster structure can be determined by comparing whether $\|\mathbf{w}^i - \mathbf{w}^j\|_2$ is below a threshold or not for any pair (i, j). Both works [33,35] decompose $\mathbf{W}$ as $\mathbf{W} = \mathbf{L}\mathbf{S}$ where columns in $\mathbf{L}$ consist of basis parameter vectors in different clusters and $\mathbf{S}$ contains combination coefficients. Both methods penalize the complexity of $\mathbf{L}$ via the squared Frobenius norm but they learn $\mathbf{S}$ in different ways. Specifically, the method in [33] aims to identify overlapping task clusters where each task can belong to multiple clusters and hence it learns a sparse $\mathbf{S}$ via the ℓ_1 regularization, while in [35], each task lies in only one cluster and hence the ℓ_2 norm of each column in the 0/1 matrix $\mathbf{S}$ is enforced to be 1.

Task-relation learning approach

In this approach, task relations are used to reflect the task relatedness and some examples for the task relations include task similarities and task covariances, just to name a few.

In earlier studies on this approach, task relations are either defined by model assumptions [36,37] or given by a priori information [38–41]. These two ways are not ideal and practical since model assumptions are hard to verify for real-world applications and a priori information is difficult to obtain. A more advanced way is to learn the task relations from data, which is the focus of this section.

A multi-task Gaussian process is proposed in [42] to define a prior on f_j^i, the functional value corresponding to $\mathbf{x}_j^i$, as $\mathbf{f} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$, where $\mathbf{f} = (f_1^1, \ldots, f_{n_m}^m)^T$. The entry in $\mathbf{\Sigma}$ corresponding to the covariance between f_j^i and f_q^p is defined as $\sigma(f_j^i, f_q^p) = \omega_{ip}k(\mathbf{x}_j^i, \mathbf{x}_q^p)$, where $k(\cdot, \cdot)$ defines a kernel function and ω_{ip} is the covariance between

tasks $\mathcal{T}_i$ and $\mathcal{T}_p$. Then, based on the Gaussian likelihood on labels given $\mathbf{f}$, the marginal likelihood, which has an analytical form, is used to learn $\mathbf{\Omega}$, the task covariance to reflect the task relatedness, with its (i, p)th entry as ω_{ip}. In order to utilize Bayesian averaging to achieve better performance, a multi-task generalized t process is proposed in [43] by placing an inverse-Wishart prior on $\mathbf{\Omega}$.

A regularized model called multi-task-relationship learning (MTRL) method is proposed in [44,45] by placing a matrix-variate normal prior on $\mathbf{W}$: $\mathbf{W} \sim \mathcal{MN}(\mathbf{0}, \mathbf{I}, \mathbf{\Omega})$, where $\mathcal{MN}(\mathbf{M}, \mathbf{A}, \mathbf{B})$ denotes a matrix-variate normal distribution with $\mathbf{M}$, $\mathbf{A}$, $\mathbf{B}$ as the mean, row covariance and column covariance. This prior corresponds to a regularizer $\mathrm{tr}(\mathbf{W}\mathbf{\Omega}^{-1}\mathbf{W}^T)$ where the PSD task covariance $\mathbf{\Omega}$ is required to satisfy $\mathrm{tr}(\mathbf{\Omega}) \leq 1$. The MTRL method is generalized to multi-task boosting [46] and multi-label learning [47], where each label is treated as a task, and extended to learn sparse task relations in [48]. A model similar to the MTRL method is proposed in [49] by assigning a prior on $\mathbf{W}$ as $\mathbf{W} \sim \mathcal{MN}(\mathbf{0}, \mathbf{\Omega}_1, \mathbf{\Omega}_2)$, and it learns the sparse inverse of $\mathbf{\Omega}_1$ and $\mathbf{\Omega}_2$. Since the prior used in the MTRL method implies that $\mathbf{W}^T\mathbf{W}$ follows a Wishart distribution as $\mathcal{W}(\mathbf{0}, \mathbf{\Omega})$, the MTRL method is generalized in [50] by studying a high-order prior: $(\mathbf{W}^T\mathbf{W})^t \sim \mathcal{W}(\mathbf{0}, \mathbf{\Omega})$, where t is a positive integer. In [51], a similar regularizer to that of the MTRL method is proposed by assuming a parametric form of $\mathbf{\Omega}$ as $\mathbf{\Omega}^{-1} = (\mathbf{I}_m - \mathbf{A})(\mathbf{I}_m - \mathbf{A})^T$, where $\mathbf{A}$ is an asymmetric task relation claimed in [51]. Unlike the aforementioned methods, which rely on global learning models, local learning methods such as the k-nearest-neighbor (kNN) classifier are extended in [52] to the multi-task setting and the learning function is defined as $f(\mathbf{x}_j^i) = \sum_{(p,q)\in N_k(i,j)} \sigma_{ip} s(\mathbf{x}_j^i, \mathbf{x}_q^p) y_q^p$, where $N_k(i, j)$ denotes the set of task and instance indices for k nearest neighbors of $\mathbf{x}_j^i$, $s(\cdot, \cdot)$ defines the similarity between instances, and σ_{ip} represents the similarity of task $\mathcal{T}_p$ to $\mathcal{T}_i$. By enforcing σ_{ip} to be close to σ_{pi}, a regularizer $\|\mathbf{\Sigma} - \mathbf{\Sigma}^T\|_F^2$ is proposed in [52] to learn task similarities, where each σ_{ip} needs to satisfy that $\sigma_{ii} \geq 0$ and $|\sigma_{ip}| \leq \sigma_{ii}$ for $i \neq p$.

Dirty approach

The dirty approach assumes the decomposition of the parameter matrix $\mathbf{W}$ as $\mathbf{W} = \mathbf{U} + \mathbf{V}$, where $\mathbf{U}$ and $\mathbf{V}$ capture different parts of the task relatedness. The objective functions of different models in this approach can be unified to minimize the training loss on all the tasks as well as two regularizers, $g(\mathbf{U})$ and $h(\mathbf{V})$, on $\mathbf{U}$ and $\mathbf{V}$, respectively. Hence, the different methods belonging to this approach differ in the choices of $g(\mathbf{U})$ and $h(\mathbf{V})$.

Here we introduce five methods in this approach, i.e. [53–57]. Different choices of $g(\mathbf{U})$ and $h(\mathbf{V})$ for the five methods are shown in Table 1. Based on Table 1, we can see that the choices of $g(\mathbf{U})$ in [53,56] make $\mathbf{U}$ row-sparse via the $\ell_{\infty,1}$ and $\ell_{2,1}$ norms, respectively. The choices of $g(\mathbf{U})$ in [54,55] enforce $\mathbf{U}$ to be low-rank via the trace norm as the regularizer and constraint, respectively. Unlike these methods, $g(\mathbf{U})$ in [57] penalizes its complexity via the squared Frobenius norm and clusters feature in different tasks based on the fused lasso regularizer. For $\mathbf{V}$, $h(\mathbf{V})$ makes it sparse via the ℓ_1 norm in [53,54] and column-sparse via the $\ell_{2,1}$ norm in [55,56], while in [57], $h(\mathbf{V})$ penalizes the complexity of $\mathbf{V}$ via the squared Frobenius norm.

In the decomposition, $\mathbf{U}$ mainly identifies the task relatedness among tasks similar to the feature selection approach or low-rank approach while $\mathbf{V}$ is capable of capturing noises or outliers via the sparsity. The combination of $\mathbf{U}$ and $\mathbf{V}$ can help the learner become more robust.

Multi-level approach

As a generalization of the dirty approach, the multi-level approach decomposes the parameter matrix $\mathbf{W}$ into h component matrices $\{\mathbf{W}_i\}_{i=1}^h$, i.e. $\mathbf{W} = \sum_{i=1}^h \mathbf{W}_i$, where the number of levels, h, is no smaller than 2. In the following, we show how the multi-level decomposition can help model complex task structures.

In the task-clustering approach, different task clusters usually have no overlap, which may restrict the expressive power of the resulting learners. In [58], all possible task clusters are enumerated,

Table 1. Choices of $g(\mathbf{U})$ and $h(\mathbf{V})$ for different methods in the dirty approach.

Method	$g(\mathbf{U})$	$h(\mathbf{V})$
[53]	$g(\mathbf{U}) = \lambda_1 \|\mathbf{U}\|_{\infty,1}$	$h(\mathbf{V}) = \lambda_2 \|\mathbf{V}\|_1$
[54]	$g(\mathbf{U}) = \begin{cases} 0, & \text{if } \|\mathbf{U}\|_{S(1)} \leq \lambda_1 \\ +\infty, & \text{otherwise.} \end{cases}$	$h(\mathbf{V}) = \lambda_2 \|\mathbf{V}\|_1$
[55]	$g(\mathbf{U}) = \lambda_1 \|\mathbf{U}\|_{S(1)}$	$h(\mathbf{V}) = \lambda_2 \|\mathbf{V}^T\|_{2,1}$
[56]	$g(\mathbf{U}) = \lambda_1 \|\mathbf{U}\|_{2,1}$	$h(\mathbf{V}) = \lambda_2 \|\mathbf{V}^T\|_{2,1}$
[57]	$g(\mathbf{U}) = \lambda_1 \sum_{i=1}^d \sum_{k>j} \|u_{ij} - u_{ik}\| + \lambda_2 \|\mathbf{U}\|_F^2$	$h(\mathbf{V}) = \lambda_3 \|\mathbf{V}\|_F^2$

leading to $2^m - 1$ task clusters, and they are organized in a tree with the root node as a dummy node, where the parent–child relation in the tree is the 'subset of' relation. This tree has 2^m nodes, each of which corresponds to a level, and hence an index t denotes both a node in the tree and the corresponding level. In order to handle a tree with such a large number of nodes, authors make an assumption that if a cluster is not useful then none of its supersets are either, which means that if a node in the tree is not helpful then none of its descendants are either. Based on this assumption, a regularizer based on the squared $\ell_{p,1}$ norm is devised, i.e. $\left(\sum_{v\in V}\lambda_v\left(\sum_{t\in D(v)} s\left(\mathbf{W}_t\right)^p\right)^{\frac{1}{p}}\right)^2$, where V denotes the set of nodes in the tree, λ_v is a regularization parameter for node v, and $D(v)$ denotes the set of descendants of v. Here $s\left(\mathbf{W}_t\right)$ uses the regularizer proposed in [36] to enforce different columns in $\mathbf{W}_t$ to be close to their average. Unlike [58] where each level involves a subset of tasks, a multi-level task-clustering method is proposed in [34] to cluster all the tasks at each level based on a structurally sparse regularizer $\sum_{i=1}^{h}\frac{\lambda}{\eta^{i-1}}\sum_{k>j}\|\mathbf{w}_i^j - \mathbf{w}_i^k\|_2$.

In [59], each component matrix is assumed to be jointly sparse and row-sparse but in different proportions, which are more similar in successive component matrices. In order to achieve this, a regularizer, i.e. $\sum_{i=1}^{h}\left(\frac{h-i}{h-1}\|\mathbf{W}_i\|_{2,1} + \frac{i-1}{h-1}\|\mathbf{W}_i\|_1\right)$, is constructed.

Unlike the aforementioned methods where different component matrices have no direct interaction, in [60], with direct connections between component matrices at successive levels, the complex hierarchical/tree structure among tasks can be learned from data. Specifically, built on the multi-level task-clustering method [34], a sequential constraint, i.e. $|\mathbf{w}_{i-1}^j - \mathbf{w}_{i-1}^k| \geq |\mathbf{w}_i^j - \mathbf{w}_i^k|\ \forall i \geq 2\ \forall k > j$, is devised in [60] to help make the whole structure become a tree.

Compared with the dirty approach that focuses on identifying noises or outliers, the multi-level approach is capable of modeling more complex task structures such as complex task clusters and tree structures.

Instance-based MTSL

There are few works in this category with the multi-task distribution matching method proposed in [61] as a representative work. Specifically, it first estimates the ratio between probabilities that each instance is from its own task and from a mixture of all the tasks. After determining ratios via softmax functions, this method uses ratios to determine the instance weights and then learns model parameters for each task based on weighted instances from all the tasks.

Discussion

Feature-based MTSL can learn a common feature representation for different tasks and it is more suitable for applications whose original feature representation is not so informative and discriminative, e.g. in computer vision, natural language processing and speech. However, feature-based MTSL can easily be affected by outlier tasks that are unrelated to other tasks, since it is difficult to learn a common feature representation for outlier tasks that are unrelated to each other. Given a good feature representation, parameter-based MTSL can learn more accurate model parameters and it is more robust to outlier tasks via a robust representation of model parameters. Hence feature-based MTSL is complemental to parameter-based MTSL. Instance-based MTSL, which is currently being explored, seems parallel to the other two categories.

In summary, the MTSL setting is the most important one in the research of MTL since it sets the stage for research in other settings. Among the existing research efforts in MTL, about 90% of works study the MTSL setting, while in the MTSL setting, the feature-based and parameter-based MTSL attract most attention from the community.

MULTI-TASK UNSUPERVISED LEARNING

Unlike multi-task supervised learning where each data instance is associated with a label, in multi-task unsupervised learning, the training set $\mathcal{D}_i$ of the ith task consists of only n_i data instances $\{\mathbf{x}_j^i\}$ and the goal of multi-task unsupervised learning is to exploit the information contained in $\mathcal{D}_i$. Typical unsupervised learning tasks include clustering, dimensionality reduction, manifold learning, visualization and so on, but multi-task unsupervised learning mainly focuses on multi-task clustering. Clustering is to divide a set of data instances into several groups, each of which has similar instances, and hence multi-task clustering aims to conduct clustering on multiple datasets by leveraging useful information contained in different datasets.

Not very many studies on multi-task clustering exist. In [62], two multi-task-clustering methods are proposed. These two methods extend the MTFL and MTRL methods [5,44], two models in the MTSL setting, to the clustering scenario and the formulations in the proposed two multi-task-clustering methods are almost identical to those in the MTFL and MTRL methods, with the only difference being

that the labels are treated as unknown cluster indicators that need to be learned from data.

MULTI-TASK SEMI-SUPERVISED LEARNING

In many applications, data usually require a great deal of manual labor to label, making labeled data not so sufficient, but in many situations, unlabeled data are ample. So in this case, unlabeled data are utilized to help improve the performance of supervised learning, leading to semi-supervised learning, whose training set consists of a mixture of labeled and unlabeled data. In multi-task semi-supervised learning, the goal is the same, where unlabeled data are used to improve the performance of supervised learning while different supervised tasks share useful information to help each other.

Based on the nature of each task, multi-task semi-supervised learning can be classified into two categories: multi-task semi-supervised classification and multi-task semi-supervised regression. For multi-task semi-supervised classification, a method proposed in [63,64] follows the task-clustering approach to do task clustering on different tasks based on a relaxed Dirichlet process, while in each task, random walk is used to exploit useful information contained in the unlabeled data. Unlike [63,64], a semi-supervised multi-task regression method is proposed in [65], where each task adopts a Gaussian process and unlabeled data are used to define the kernel function, and Gaussian processes in all the tasks share a common prior on kernel parameters.

MULTI-TASK ACTIVE LEARNING

The setting of multi-task active learning, where each task has a small number of labeled data and a large amount of unlabeled data in the training set, is almost identical to that of multi-task semi-supervised learning. However, unlike multi-task semi-supervised learning, which exploits information contained in the unlabeled data, in multi-task active learning, each task selects informative unlabeled data to query an oracle to actively acquire their labels. Hence the criterion for the selection of unlabeled data is the main research focus in multi-task active learning [66–68].

Specifically, two criteria are proposed in [66] to make sure that the selected unlabeled instances are informative for all the tasks instead of only one task. Unlike [66], in [67] where the learner in each task is a supervised latent Dirichlet allocation model, the selection criterion for unlabeled data is the expected error reduction. Moreover, a selection strategy, a tradeoff between the learning risk of a low-rank MTL model based on the trace-norm regularization and a confidence bound similar to multi-armed bandits, is proposed in [68].

MULTI-TASK REINFORCEMENT LEARNING

Inspired by behaviorist psychology, reinforcement learning studies how to take actions in an environment to maximize the cumulative reward and it shows good performance in many applications with AlphaGo, which beats humans in the Go game, as a representative application. When environments are similar, different reinforcement learning tasks can use similar policies to make decisions, which is a motivation of the proposal of multi-task reinforcement learning [69–73].

Specifically, in [69], each reinforcement learning task is modeled by a Markov decision process (MDP) and different MDPs in all the tasks are related via a hierarchical Bayesian infinite mixture model. In [70], each task is characterized via a regionalized policy and a Dirichlet process is used to cluster tasks. In [71], the reinforcement learning model for each task is a Gaussian process temporal-difference value function model and a hierarchical Bayesian model relates value functions of different tasks. In [72], the value functions in different tasks are assumed to share sparse parameters and it applies the multi-task feature selection method with the $\ell_{2,1}$ regularization [8] and the MTFL method [5] to learn all the value functions simultaneously. In [73], an actor–mimic method, which is a combination of deep reinforcement learning and model compression techniques, is proposed to learn policy networks for multiple tasks.

MULTI-TASK ONLINE LEARNING

When the training data in multiple tasks come in a sequential way, traditional MTL models cannot handle them but multi-task online learning is capable of doing this job, as shown in some representative works [74–79].

Specifically, in [74,75], where different tasks are assumed to have a common goal, a global loss function, a combination of individual losses on each task, measures the relations between tasks, and by using absolute norms for the global loss function, several online MTL algorithms are proposed. In [76], the proposed online MTL algorithms model task relations by placing constraints on actions taken for all the tasks. In [77], online MTL algorithms, which adopt perceptrons as a basic model and measure task relations based on shared geometric structures

among tasks, are proposed for multi-task classification problems. In [78], a Bayesian online algorithm is proposed for a multi-task Gaussian process that shares kernel parameters among tasks. In [79], an online algorithm is proposed for the MTRL method [44] by updating model parameters and task covariance together.

MULTI-TASK MULTI-VIEW LEARNING

In some applications such as computer vision, each data point can be described by different feature representations; one example is image data, whose features include SIFT and wavelet, to name just a few. In this case, each feature representation is called a view and multi-view learning, a learning paradigm in machine learning, is proposed to handle such data with multiple views. Similar to supervised learning, each multi-view data point is usually associated with a label. Multi-view learning aims to exploit useful information contained in multiple views to further improve the performance over supervised learning, which can be considered as a single-view learning paradigm. As a multi-task extension of multi-view learning, multi-task multi-view learning [80,81] hopes to exploit multiple multi-view learning problems to improve the performance over each multi-view learning problem by leveraging useful information contained in related tasks.

Specifically, in [80], the first multi-task multi-view classifier is proposed to utilize the task relatedness based on common views shared by tasks and view consistency among views in each task. In [81], different views in each task achieve consensus on unlabeled data and different tasks are learned by exploiting a priori information as in [38] or learning task relations as the MTRL method did.

PARALLEL AND DISTRIBUTED MTL

When the number of tasks is large, if we directly apply a multi-task learner, the computational complexity may be high. Nowadays the computational capacity of a computer is very powerful due to the multi-CPU or multi-GPU architecture involved. So we can make use of these powerful computing facilities to devise parallel MTL algorithms to accelerate the training process. In [82], a parallel MTL method is devised to solve a subproblem of the MTRL model [44], which also occurs in many regularized methods belonging to the task-relation learning approach. Specifically, this method utilizes the FISTA algorithm to design a decomposable surrogate function with respect to all the tasks and this surrogate function can be parallelized to speed up the learning process. Moreover, three loss functions, including the hinge, ϵ-insensitive and square losses, are studied in [82], making this parallel method applicable to both classification and regression problems in MTSL.

In some cases, training data for different tasks may exist in different machines, which makes it difficult for conventional MTL models to work, even though all the training data can be moved to one machine, which incurs additional transmission and storage costs. A better option is to devise distributed MTL models that can directly operate on data distributed on multiple machines. In [83], a distributed algorithm is proposed based on a debiased lasso model and by learning one task in a machine, this algorithm achieves efficient communications.

APPLICATIONS OF MULTI-TASK LEARNING

Several areas, including computer vision, bioinformatics, health informatics, speech, natural language processing, web applications and ubiquitous computing, use MTL to boost the performance of their respective applications. In this section, we review some related works.

Computer vision

The applications of MTL in computer vision can be divided into two categories, including image-based and video-based applications.

Image-based MTL applications include two subcategories: facial images and non-facial images. Specifically, applications of MTL based on facial images include face verification [84], personalized age estimation [85], multi-cue face recognition [86], head-pose estimation [22,87], facial landmark detection [18], and facial image rotation [88]. Applications of MTL based on non-facial images include object categorization [86], image segmentation [89,90], identifying brain imaging predictors [91], saliency detection [92], action recognition [93], scene classification [94], multi-attribute prediction [95], multi-camera person re-identification [96], and immediacy prediction [97].

Applications of MTL based on videos include visual tracking [98–100] and thumbnail selection [19].

Bioinformatics and health informatics

Applications of MTL in bioinformatics and health informatics include organism modeling [101], mechanism identification of response to therapeutic targets [102], cross-platform siRNA efficacy prediction [103], detection of causal genetic markers through association analysis of multiple

populations [104], construction of personalized brain–computer interfaces [105], MHC-I binding prediction [106], splice-site prediction [106], protein subcellular location prediction [107], Alzheimer's disease assessment scale cognitive subscale [108], prediction of cognitive outcomes from neuroimaging measures in Alzheimer's disease [109], identification of longitudinal phenotypic markers for Alzheimer's disease progression prediction [110], prioritization of disease genes [111], biological image analysis based on natural images [20], survival analysis [112], and multiple genetic trait prediction [113].

Speech and natural language processing

Applications of MTL in speech include speech synthesis [114,115] and those for natural language processing include joint learning of six NLP tasks (i.e. part-of-speech tagging, chunking, named entity recognition, semantic role labeling, language modeling and semantically related words) [116], multi-domain sentiment classification [117], multi-domain dialog state tracking [21], machine translation [118], syntactic parsing [118], and microblog analysis [119,120].

Web applications

Web applications based on MTL include learning to rank in web searches [121], web search ranking [122], multi-domain collaborative filtering [123], behavioral targeting [124], and conversion maximization in display advertising [125].

Ubiquitous computing

Applications of MTL in ubiquitous computing include stock prediction [126], multi-device localization [127], the inverse dynamics problem for robotics [128,129], estimation of travel costs on road networks [130], travel-time prediction on road networks [131], and traffic-sign recognition [132].

THEORETICAL ANALYSIS

Learning theory, an area in machine learning, studies the theoretical aspect of learning models including MTL models. In the following, we introduce some representative works.

The theoretical analysis in MTL mainly focuses on deriving the generalization bound of MTL models. It is well known that the generalization performance of MTL models on unseen test data is the main concern in MTL and machine learning. However, since the underlying data distribution is difficult to model, the generalization performance cannot be computed and instead the generalization bound is used to provide an upper bound for the generalization performance.

The first generalization bound for MTL is derived in [133] for a general MTL model. Then there are many studies to analyze generalization bounds of different MTL approaches, including e.g. [7,134] for the feature transform approach, [135] for the feature selection approach, [24,135–138] for the low-rank approach, [136] for the task-relation learning approach, and [138] for the dirty approach.

CONCLUSIONS

In this paper, we give an overview of MTL. Firstly, we give a definition of MTL. After that, different settings of MTL are presented, including multi-task supervised learning, multi-task unsupervised learning, multi-task semi-supervised learning, multi-task active learning, multi-task reinforcement learning, multi-task online learning and multi-task multi-view learning. For each setting, we introduce its representative models. Then parallel and distributed MTL models, which can help speed up the learning process, are discussed. Finally, we review the applications of MTL in various areas and present theoretical analyses for MTL.

Recently deep learning has become popular in many applications and several deep models have been devised for MTL. Almost all the deep models just share hidden layers for different tasks; this way of sharing knowledge among tasks is very useful when all the tasks are very similar, but when this assumption is violated, the performance will significantly deteriorate. We think one future direction for multi-task deep models is to design more flexible architectures that can tolerate dissimilar tasks and even outlier tasks. Moreover, the deep-learning, task-clustering and multi-level approaches lack theoretical foundations and more analyses are needed to guide the research in these approaches.

FUNDING

This work was supported by the National Basic Research Program of China (973 Program) (2014CB340304), the Hong Kong CERG projects (16211214, 16209715 and 16244616), the National Natural Science Foundation of China (61473087 and 61673202), and the Natural Science Foundation of Jiangsu Province (BK20141340).

REFERENCES

1. Caruana R. Multitask learning. *Mach Learn* 1997; **28**: 41–75.
2. Pan SJ and Yang Q. A survey on transfer learning. *IEEE Trans Knowl Data Eng* 2010; **22**: 1345–59.

3. Zhang M and Zhou Z. A review on multi-label learning algorithms. *IEEE Trans Knowl Data Eng* 2014; **26**: 1819–37.
4. Zhang Y and Yang Q. *A survey on multi-task learning*. arXiv:1707.08114.
5. Argyriou A, Evgeniou T and Pontil M. Multi-task feature learning. In: *Advances in Neural Information Processing Systems 19*. 2006, 41–8.
6. Argyriou A, Evgeniou T and Pontil M. Convex multi-task feature learning. *Mach Learn* 2008; **73**: 243–72.
7. Maurer A, Pontil M and Romera-Paredes B. Sparse coding for multitask and transfer learning. In: *Proceedings of the 30th International Conference on Machine Learning*. 2013, 343–51.
8. Obozinski G, Taskar B and Jordan M. Multi-task feature selection. *Ph.D. Thesis*. University of California, Berkeley Department of Statistics 2006.
9. Obozinski G, Taskar B and Jordan M. Joint covariate selection and joint subspace selection for multiple classification problems. *Stat Comput* 2010; **20**: 231–52.
10. Liu H, Palatucci M and Zhang J. Blockwise coordinate descent procedures for the multi-task lasso, with applications to neural semantic basis discovery. In: *Proceedings of the 26th International Conference on Machine Learning*. 2009, 649–56.
11. Gong P, Ye J and Zhang C. Multi-stage multi-task feature learning. *J Mach Learn Res* 2013; **14**: 2979–3010.
12. Lozano AC and Swirszcz G. Multi-level lasso for sparse multi-task regression. In: *Proceedings of the 29th International Conference on Machine Learning*. 2012.
13. Wang X, Bi J and Yu S *et al*. On multiplicative multitask feature learning. In: *Advances in Neural Information Processing Systems 27*. 2014, 2411–9.
14. Han L, Zhang Y and Song G *et al*. Encoding tree sparsity in multi-task learning: a probabilistic framework. In: *Proceedings of the 28th AAAI Conference on Artificial Intelligence*. 2014, 1854–60.
15. Zhang Y, Yeung DY and Xu Q. Probabilistic multi-task feature selection. In: *Advances in Neural Information Processing Systems 23*. 2010, 2559–67.
16. Hernández-Lobato D and Hernández-Lobato JM. Learning feature selection dependencies in multi-task learning. In: *Advances in Neural Information Processing Systems 26*. 2013, 746–54.
17. Hernández-Lobato D, Hernández-Lobato JM and Ghahramani Z. A probabilistic model for dirty multi-task feature selection. In: *Proceedings of the 32nd International Conference on Machine Learning*. 2015, 1073–82.
18. Zhang Z, Luo P and Loy CC *et al*. Facial landmark detection by deep multi-task learning. In: *Proceedings of the 13th European Conference on Computer Vision*. 2014, 94–108.
19. Liu W, Mei T and Zhang Y *et al*. Multi-task deep visual-semantic embedding for video thumbnail selection. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 2015, 3707–15.
20. Zhang W, Li R and Zeng T *et al*. 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*. 2015, 1475–84.
21. Mrksic N, Séaghdha DÓ and Thomson B *et al*. Multi-domain dialog state tracking using recurrent neural networks. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*. 2015, 794–9.
22. Li S, Liu Z and Chan AB. Heterogeneous multi-task learning for human pose estimation with deep convolutional neural network. *Int J Comput Vis* 2015; **113**: 19–36.
23. Misra I, Shrivastava A and Gupta A *et al*. Cross-stitch networks for multi-task learning. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 2016, 3994–4003.
24. Ando RK and Zhang T. A framework for learning predictive structures from multiple tasks and unlabeled data. *J Mach Learn Res* 2005; **6**: 1817–53.
25. Chen J, Tang L and Liu J *et al*. A convex formulation for learning shared structures from multiple tasks. In: *Proceedings of the 26th International Conference on Machine Learning*. 2009, 137–44.
26. Pong TK, Tseng P and Ji S *et al*. Trace norm regularization: reformulations, algorithms, and multi-task learning. *SIAM J Optim* 2010; **20**: 3465–89.
27. Han L and Zhang Y. Multi-stage multi-task learning with reduced rank. In: *Proceedings of the 30th AAAI Conference on Artificial Intelligence*. 2016.
28. Thrun S and O'Sullivan J. Discovering structure in multiple learning tasks: the TC algorithm. In: *Proceedings of the 13th International Conference on Machine Learning*. 1996, 489–97.
29. Bakker B and Heskes T. Task clustering and gating for Bayesian multitask learning. *J Mach Learn Res* 2003; **4**: 83–99.
30. Xue Y, Liao X and Carin L *et al*. Multi-task learning for classification with Dirichlet process priors. *J Mach Learn Res* 2007; **8**: 35–63.
31. Jacob L, Bach FR and Vert JP. Clustered multi-task learning: a convex formulation. In: *Advances in Neural Information Processing Systems 21*. 2008, 745–52.
32. Kang Z, Grauman K and Sha F. Learning with whom to share in multi-task feature learning. In: *Proceedings of the 28th International Conference on Machine Learning*. 2011, 521–8.
33. Kumar A and III HD. Learning task grouping and overlap in multi-task learning. In: *Proceedings of the 29th International Conference on Machine Learning*. 2012.
34. Han L and Zhang Y. Learning multi-level task groups in multi-task learning. In: *Proceedings of the 29th AAAI Conference on Artificial Intelligence*. 2015.
35. Barzilai A and Crammer K. Convex multi-task learning by clustering. In: *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*. 2015.
36. Evgeniou T and Pontil M. Regularized multi-task learning. In: *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2004, 109–17.
37. Parameswaran S and Weinberger KQ. Large margin multi-task metric learning. In: *Advances in Neural Information Processing Systems 23*. 2010, 1867–75.
38. Evgeniou T, Micchelli CA and Pontil M. Learning multiple tasks with kernel methods. *J Mach Learn Res* 2005; **6**: 615–37.
39. Kato T, Kashima H and Sugiyama M *et al*. Multi-task learning via conic programming. In: *Advances in Neural Information Processing Systems 20*. 2007, 737–44.
40. Kato T, Kashima H and Sugiyama M *et al*. Conic programming for multitask learning. *IEEE Trans Knowl Data Eng* 2010; **22**: 957–68.
41. Görnitz N, Widmer C and Zeller G *et al*. Hierarchical multitask structured output learning for large-scale sequence segmentation. In: *Advances in Neural Information Processing Systems 24*. 2011, 2690–8.
42. Bonilla EV, Chai KMA and Williams CKI. Multi-task Gaussian process prediction. In: *Advances in Neural Information Processing Systems 20*. 2007, 153–60.
43. Zhang Y and Yeung DY. Multi-task learning using generalized *t* process. In: *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*. 2010, 964–71.
44. Zhang Y and Yeung DY. A convex formulation for learning task relationships in multi-task learning. In: *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*. 2010, 733–42.

45. Zhang Y and Yeung DY. A regularization approach to learning task relationships in multitask learning. *ACM Trans Knowl Discov Data* 2014; **8**: 12.
46. Zhang Y and Yeung DY. Multi-task boosting by exploiting task relationships. In: *Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*. 2012, 697–710.
47. Zhang Y and Yeung DY. Multilabel relationship learning. *ACM Trans Knowl Discov Data* 2013; **7**: 7.
48. Zhang Y and Yang Q. Learning sparse task relations in multi-task learning. In: *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. 2017.
49. Zhang Y and Schneider JG. Learning multiple tasks with a sparse matrix-normal penalty. In: *Advances in Neural Information Processing Systems 23*. 2010, 2550–8.
50. Zhang Y and Yeung DY. Learning high-order task relationships in multi-task learning. In: *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*. 2013.
51. Lee G, Yang E and Hwang SJ. Asymmetric multi-task learning based on task relatedness and loss. In: *Proceedings of the 33rd International Conference on Machine Learning*. 2016, 230–8.
52. Zhang Y. Heterogeneous-neighborhood-based multi-task local learning algorithms. In: *Advances in Neural Information Processing Systems 26*. 2013.
53. Jalali A, Ravikumar P and Sanghavi S *et al*. A dirty model for multi-task learning. In: *Advances in Neural Information Processing Systems 23*. 2010, 964–72.
54. Chen J, Liu J and Ye J. Learning incoherent sparse and low-rank patterns from multiple tasks. In: *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2010, 1179–88.
55. Chen J, Zhou J and 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*. 2011, 42–50.
56. Gong P, Ye J and Zhang C. Robust multi-task feature learning. In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2012, 895–903.
57. Zhong W and Kwok JT. Convex multitask learning with flexible task clusters. In: *Proceedings of the 29th International Conference on Machine Learning*. 2012.
58. Jawanpuria P and Nath JS. A convex feature learning formulation for latent task structure discovery. In: *Proceedings of the 29th International Conference on Machine Learning*. 2012.
59. Zweig A and Weinshall D. Hierarchical regularization cascade for joint learning. In: *Proceedings of the 30th International Conference on Machine Learning*. 2013, 37–45.
60. Han L and Zhang Y. Learning tree structure in multi-task learning. In: *Proceedings of the 21st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2015.
61. Bickel S, Bogojeska J and Lengauer T *et al*. Multi-task learning for HIV therapy screening. In: *Proceedings of the Twenty-Fifth International Conference on Machine Learning*. 2008, 56–63.
62. Zhang X. Convex discriminative multitask clustering. *IEEE Trans Pattern Anal Mach Intell* 2015; **37**: 28–40.
63. Liu Q, Liao X and Carin L. Semi-supervised multitask learning. In: *Advances in Neural Information Processing Systems 20*. 2007, 937–44.
64. Liu Q, Liao X and Li H *et al*. Semisupervised multitask learning. *IEEE Trans Pattern Anal Mach Intell* 2009; **31**: 1074–86.
65. Zhang Y and Yeung D. Semi-supervised multi-task regression. In: *Proceedings of European Conference on Machine Learning and Knowledge Discovery in Databases*. 2009, 617–31.
66. Reichart R, Tomanek K and Hahn U *et al*. Multi-task active learning for linguistic annotations. In: *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*. 2008, 861–9.
67. Acharya A, Mooney RJ and Ghosh J. Active multitask learning using both latent and supervised shared topics. In: *Proceedings of the 2014 SIAM International Conference on Data Mining*. 2014, 190–8.
68. Fang M and Tao D. Active multi-task learning via bandits. In: *Proceedings of the 2015 SIAM International Conference on Data Mining*. 2015, 505–13.
69. Wilson A, Fern A and Ray S *et al*. Multi-task reinforcement learning: a hierarchical Bayesian approach. In: *Proceedings of the Twenty-Fourth International Conference on Machine Learning*. 2007, 1015–22.
70. Li H, Liao X and Carin L. Multi-task reinforcement learning in partially observable stochastic environments. *J Mach Learn Res* 2009; **10**: 1131–86.
71. Lazaric A and Ghavamzadeh M. Bayesian multi-task reinforcement learning. In: *Proceedings of the 27th International Conference on Machine Learning*. 2010, 599–606.
72. Calandriello D, Lazaric A and Restelli M. Sparse multi-task reinforcement learning. In: *Advances in Neural Information Processing Systems 27*. 2014, 819–27.
73. Parisotto E, Ba J and Salakhutdinov R. Actor-mimic: deep multitask and transfer reinforcement learning. In: *Proceedings of the 4th International Conference on Learning Representations*. 2016.
74. Dekel O, Long PM and Singer Y. Online multitask learning. In: *Proceedings of the 19th Annual Conference on Learning Theory*. 2006, 453–67.
75. Dekel O, Long PM and Singer Y. Online learning of multiple tasks with a shared loss. *J Mach Learn Res* 2007; **8**: 2233–64.
76. Lugosi G, Papaspiliopoulos O and Stoltz G. Online multi-task learning with hard constraints. In: *Proceedings of the 22nd Conference on Learning Theory*. 2009.
77. Cavallanti G, Cesa-Bianchi N and Gentile C. Linear algorithms for online multitask classification. *J Mach Learn Res* 2010; **11**: 2901–34.
78. Pillonetto G, Dinuzzo F and Nicolao GD. Bayesian online multitask learning of Gaussian processes. *IEEE Trans Pattern Anal Mach Intell* 2010; **32**: 193–205.
79. Saha A, Rai P, Daumé H and Venkatasubramanian S. Online learning of multiple tasks and their relationships. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. 2011, 643–51.
80. He J and Lawrence R. A graph-based framework for multi-task multi-view learning. In: *Proceedings of the 28th International Conference on Machine Learning*. 2011, 25–32.
81. Zhang J and Huan J. Inductive multi-task learning with multiple view data. In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2012, 543–51.
82. Zhang Y. Parallel multi-task learning. In: *Proceedings of the IEEE International Conference on Data Mining*. 2015.
83. Wang J, Kolar M and Srebro N. Distributed multi-task learning. In: *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*. 2016, 751–60.
84. Wang X, Zhang C and Zhang Z. Boosted multi-task learning for face verification with applications to web image and video search. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 2009, 142–9.
85. Zhang Y and Yeung DY. Multi-task warped Gaussian process for personalized age estimation. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 2010.

86. Yuan X and Yan S. Visual classification with multi-task joint sparse representation. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 2010, 3493–500.
87. Yan Y, Ricci E and Ramanathan S *et al*. No matter where you are: flexible graph-guided multi-task learning for multi-view head pose classification under target motion. In: *Proceedings of IEEE International Conference on Computer Vision*. 2013, 1177–84.
88. Yim J, Jung H and Yoo B *et al*. Rotating your face using multi-task deep neural network. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 2015, 676–84.
89. An Q, Wang C and Shterev I *et al*. Hierarchical kernel stick-breaking process for multi-task image analysis. In: *Proceedings of the 25th International Conference on Machine Learning*. 2008, 17–24.
90. Cheng B, Liu G and Wang J *et al*. Multi-task low-rank affinity pursuit for image segmentation. In: *Proceedings of IEEE International Conference on Computer Vision*. 2011, 2439–46.
91. Wang H, Nie F and Huang H *et al*. Sparse multi-task regression and feature selection to identify brain imaging predictors for memory performance. In: *Proceedings of IEEE International Conference on Computer Vision*. 2011, 557–62.
92. Lang C, Liu G and Yu J *et al*. Saliency detection by multitask sparsity pursuit. *IEEE Trans Image Process* 2012; **21**: 1327–38.
93. Yuan C, Hu W and Tian G *et al*. Multi-task sparse learning with beta process prior for action recognition. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 2013, 423–9.
94. Lapin M, Schiele B and Hein M. Scalable multitask representation learning for scene classification. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 2014, 1434–41.
95. Abdulnabi AH, Wang G and Lu J *et al*. Multi-task CNN model for attribute prediction. *IEEE Trans Multimed* 2015; **17**: 1949–59.
96. Su C, Yang F and Zhang S *et al*. Multi-task learning with low rank attribute embedding for person re-identification. In: *Proceedings of IEEE International Conference on Computer Vision*. 2015, 3739–47.
97. Chu X, Ouyang W and Yang W *et al*. Multi-task recurrent neural network for immediacy prediction. In: *Proceedings of IEEE International Conference on Computer Vision*. 2015, 3352–60.
98. Zhang T, Ghanem B and Liu S *et al*. Robust visual tracking via multi-task sparse learning. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 2012, 2042–9.
99. Zhang T, Ghanem B and Liu S *et al*. Robust visual tracking via structured multi-task sparse learning. *Int J Comput Vis* 2013; **101**: 367–83.
100. Hong Z, Mei X and Prokhorov DV *et al*. Tracking via robust multi-task multi-view joint sparse representation. In: *Proceedings of IEEE International Conference on Computer Vision*. 2013, 649–56.
101. Widmer C, Leiva J and Altun Y *et al*. Leveraging sequence classification by taxonomy-based multitask learning. In: *Proceedings of the 14th Annual International Conference on Research in Computational Molecular Biology*. 2010, 522–34.
102. Zhang K, Gray JW and Parvin B. Sparse multitask regression for identifying common mechanism of response to therapeutic targets. *Bioinformatics* 2010; **26**: 97–105.
103. Liu Q, Xu Q and Zheng VW *et al*. Multi-task learning for cross-platform siRNA efficacy prediction: an in-silico study. *BMC Bioinformatics* 2010; **11**: 181.
104. Puniyani K, Kim S and Xing EP. Multi-population GWA mapping via multi-task regularized regression. *Bioinformatics* 2010; **26**: 208–16.
105. Alamgir M, Grosse-Wentrup M and Altun Y. Multitask learning for brain-computer interfaces. In: *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*. 2010, 17–24.
106. Widmer C, Toussaint NC and Altun Y *et al*. Inferring latent task structure for multitask learning by multiple kernel learning. *BMC Bioinformatics* 2010; **11**: S5.
107. Xu Q, Pan SJ and Xue HH *et al*. Multitask learning for protein subcellular location prediction. *IEEE ACM Trans Comput Biol Bioinformatics* 2011; **8**: 748–59.
108. Zhou J, Yuan L and Liu J *et al*. A multi-task learning formulation for predicting disease progression. In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2011, 814–22.
109. Wan J, Zhang Z and Yan J *et al*. Sparse Bayesian multi-task learning for predicting cognitive outcomes from neuroimaging measures in Alzheimer's disease. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 2012, 940–7.
110. Wang H, Nie F and Huang H *et al*. High-order multi-task feature learning to identify longitudinal phenotypic markers for alzheimer's disease progression prediction. In: *Advances in Neural Information Processing Systems 25*. 2012, 1286–94.
111. Mordelet F and Vert J. ProDiGe: Prioritization of disease genes with multitask machine learning from positive and unlabeled examples. *BMC Bioinformatics* 2011; **12**: 389.
112. Li Y, Wang J and Ye J *et al*. A multi-task learning formulation for survival analysis. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, 1715–24.
113. He D, Kuhn D and Parida L. Novel applications of multitask learning and multiple output regression to multiple genetic trait prediction. *Bioinformatics* 2016; **32**: 37–43.
114. Wu Z, Valentini-Botinhao C and Watts O *et al*. Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis. In: *Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2015, 4460–4.
115. Hu Q, Wu Z and Richmond K *et al*. Fusion of multiple parameterisations for DNN-based sinusoidal speech synthesis with multi-task learning. In: *Proceedings of the 16th Annual Conference of the International Speech Communication Association*. 2015, 854–8.
116. Collobert R and 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*. 2008, 160–7.
117. Wu F and Huang Y. Collaborative multi-domain sentiment classification. In: *Proceedings of the 2015 IEEE International Conference on Data Mining*. 2015, 459–68.
118. Luong M, Le QV and Sutskever I *et al*. Multi-task sequence to sequence learning. In: *Proceedings of the 4th International Conference on Learning Representations*. 2016.
119. Zhao L, Sun Q and Ye J *et al*. Multi-task learning for spatio-temporal event forecasting. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2015, 1503–12.
120. Zhao L, Sun Q and Ye J *et al*. Feature constrained multi-task learning models for spatiotemporal event forecasting. *IEEE Trans Knowl Data Eng* 2017; **29**: 1059–72.
121. Bai J, Zhou K and Xue G *et al*. Multi-task learning for learning to rank in web search. In: *Proceedings of the 18th ACM Conference on Information and Knowledge Management*. 2009, 1549–52.
122. Chapelle O, Shivaswamy PK and Vadrevu S *et al*. Multi-task learning for boosting with application to web search ranking. In: *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2010, 1189–98.

123. Zhang Y, Cao B and Yeung DY. Multi-domain collaborative filtering. In: *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*. 2010, 725–32.
124. Ahmed A, Aly M and Das A *et al*. Web-scale multi-task feature selection for behavioral targeting. In: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. 2012, 1737–41.
125. Ahmed A, Das A and Smola AJ. Scalable hierarchical multitask learning algorithms for conversion optimization in display advertising. In: *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*. 2014, 153–62.
126. Ghosn J and Bengio Y. Multi-task learning for stock selection. In: *Advances in Neural Information Processing Systems 9*. 1996, 946–52.
127. Zheng VW, Pan SJ and Yang Q *et al*. Transferring multi-device localization models using latent multi-task learning. In: *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*. 2008, 1427–32.
128. Chai KMA, Williams CKI and Klanke S *et al*. Multi-task Gaussian process learning of robot inverse dynamics. In: *Advances in Neural Information Processing Systems 21, December 8-11, 2008*. 2008, 265–72.
129. Yeung DY and Zhang Y. Learning inverse dynamics by Gaussian process regression under the multi-task learning framework. In: *The Path to Autonomous Robots*. Berlin: Springer, 2009, 131–42.
130. Zheng J and Ni LM. Time-dependent trajectory regression on road networks via multi-task learning. In: *Proceedings of the 27th AAAI Conference on Artificial Intelligence*. 2013.
131. Huang A, Xu L and Li Y *et al*. Robust dynamic trajectory regression on road networks: a multi-task learning framework. In: *Proceedings of IEEE International Conference on Data Mining*. 2014, 857–62.
132. Lu X, Wang Y and Zhou X *et al*. Traffic sign recognition via multi-modal tree-structure embedded multi-task learning. *IEEE Trans Intell Transport Syst* 2017; **18**: 960–72.
133. Baxter J. A model of inductive bias learning. *J Artif Intell Res* 2000; **12**: 149–98.
134. Maurer A. Bounds for linear multi-task learning. *J Mach Learn Res* 2006; **7**: 117–39.
135. Kakade SM, Shalev-Shwartz S and Tewari A. Regularization techniques for learning with matrices. *J Mach Learn Res* 2012; **13**: 1865–90.
136. Maurer A. The Rademacher complexity of linear transformation classes. In: *Proceedings of the 19th Annual Conference on Learning Theory*. 2006, 65–78.
137. Pontil M and Maurer A. Excess risk bounds for multitask learning with trace norm regularization. In: *Proceedings of the 26th Annual Conference on Learning Theory*. 2013, 55–76.
138. Zhang Y. Multi-task learning and algorithmic stability. In: *Proceedings of the 29th AAAI Conference on Artificial Intelligence*. 2015.