
Learning Multiple Tasks with Deep Relationship Networks

Mingsheng Long^{†‡} and Jianmin Wang[‡]

[†]Department of Computer Science, University of California, Berkeley 94704

[‡]School of Software, Institute for Data Science, Tsinghua University, Beijing 100084, China
mingsheng@tsinghua.edu.cn, jimwang@tsinghua.edu.cn

Abstract

Deep neural networks trained on large-scale dataset can learn transferable features that promote learning multiple tasks for inductive transfer and labeling mitigation. As deep features eventually transition from general to specific along the network, a fundamental problem is how to exploit the relationship structure across different tasks while accounting for the feature transferability in the task-specific layers. In this work, we propose a novel Deep Relationship Network (DRN) architecture for multi-task learning by discovering correlated tasks based on multiple task-specific layers of a deep convolutional neural network. DRN models the task relationship by imposing matrix normal priors over the network parameters of all task-specific layers, including higher feature layers and classifier layer that are not transferable safely. By jointly learning the transferable features and task relationships, DRN is able to alleviate the dilemma of negative-transfer in the feature layers and under-transfer in the classifier layer. Empirical evidence shows that DRN yields state-of-the-art classification results on standard multi-domain object recognition datasets.

1 Introduction

Supervised learning machines trained with limited labeled samples will be prone to overfitting, while manual labeling of sufficient training data for emerging application domains is usually prohibitive. Therefore, it is imperative to establish effective algorithms for reducing the labeling cost, typically by leveraging off-the-shelf labeled data from relevant learning tasks. Multi-task learning is based on the idea that the performance can be improved using related tasks as an inductive bias [1]. Knowing the task relationship should enable the transfer of shared knowledge from relevant tasks so that only task-specific features need to be learned. This fundamental idea has motivated a variety of methods, including multi-task feature learning that learns a shared feature representation [2, 3, 4, 5, 6], and multi-task relationship learning that models the inherent task relationship [7, 8, 9, 10, 11, 12, 13, 14].

Learning the inherent task relatedness is a hard problem, since the training data of different tasks may be sampled from different probability distributions, and may be fitted by different inductive models. In the absence of prior knowledge on the task relatedness, the distribution shift may pose a major bottleneck in transferring knowledge across different tasks. Unfortunately, if cross-task knowledge transfer is impossible, then we will overfit each task due to limited amount of labeled data. One way to circumvent this dilemma is to use an external data source, e.g. ImageNet, to extract transferable features through which the shift in the inductive biases can be reduced so that different tasks can be correlated more effectively. This idea has motivated some latest deep learning methods for learning multiple tasks [15, 16, 17, 18], which learn a shared representation in the feature layers and multiple independent classifiers in the classifier layer but without inferring the task relationships. However, this may result in *under-transfer* in the classifier layer as knowledge can not be transferred across different classifiers. Furthermore, the literature’s latest findings reveal that deep features eventually transition from general to specific along the network, and feature transferability drops significantly

in higher layers with increasing task discrepancy [19, 20], hence the sharing of all feature layers may be risky to *negative-transfer*. It remains an open problem how to exploit the task relationship across different tasks while accounting for the feature transferability in task-specific layers of the network.

In this work, we propose a new Deep Relationship Network (DRN) architecture for learning multiple tasks, which discovers the inherent task relationship based on multiple task-specific layers of a deep convolutional neural network. DRN models the task relationship by imposing matrix normal priors [21] over network parameters of all task-specific layers, including higher feature layers and classifier layer that are not transferable safely. In each layer, the task relationship is shared by all categories, which can naturally handle multi-class problems and can learn the task relationship more accurately when category-wise data is scarce. We further impose a Gaussian prior with a task mean parameter to capture the common task component, which is particularly effective for those transferable lower layers of the deep networks. By jointly learning the transferable features and the task relationships, DRN is able to alleviate the dilemma of negative-transfer in the feature layers and under-transfer in the classifier layer. Since deep models pre-trained with large-scale repositories such as ImageNet are representative for general-purpose tasks [19, 22], the proposed DRN model is trained by fine-tuning from the AlexNet model pre-trained on ImageNet [23]. Extensive empirical study shows that DRN yields state-of-the-art classification results on the standard multi-domain object recognition datasets.

2 Related Work

Multi-task learning (MTL) [1] is an important learning paradigm that jointly learns multiple tasks by exploiting shared structural representation and improves generalization by using related tasks as an inductive bias. Multi-task learning is to mitigate the effort of manual labeling for machine learning, computer vision, natural language processing, and computational biology. There are generally two categories of approaches to multi-task learning: multi-task feature learning, which learns a shared feature representation such that the dataset bias across different tasks can be reduced [2, 3, 4] or the outlier tasks can be identified [5, 6]; and multi-task relationship learning, which explicitly models the task relationship in the forms of task grouping [7, 8, 9] or task covariance [10, 11, 12, 13, 14]. While these methods have been shown to produce state-of-the-art performance, they may be restricted by their shallow learning architecture that cannot suppress task-specific variations for task correlation.

Deep neural networks learn nonlinear representations that disentangle and hide different explanatory factors of variation behind samples [24, 23]. The learned deep representations can manifest invariant factors underlying different populations and are transferable across similar tasks [19]. Hence, deep neural networks have been explored for domain adaptation [25, 20], multimodal learning [26, 27] and multi-task learning [15, 16, 17, 18], where significant performance gains have been obtained. Based on the assumption that deep neural networks can learn transferable representations across different tasks, most prior multi-task deep learning methods for natural language processing [15] and computer vision [17, 18] learn a shared representation in the feature layers and multiple independent classifiers in the classifier layer without inferring the task relationships. However, this may result in *under-transfer* in the classifier layer as knowledge can not be adaptively propagated across different classifiers. To address this issue, Srivastava et al. [16] proposed tree-based priors to the weights in the classifier layer, which learns to organize the classes into a tree hierarchy and transfers knowledge to infrequent classes for label-scarcity mitigation. While their method is specifically designed for multi-class classification instead of multi-task learning, the sharing of all feature layers may still be vulnerable to *negative-transfer*, as the higher layers of deep networks are tailored to fit specific tasks and may not be safely transferable [19]. We propose a deep relationship network that jointly learns transferable features and task relationships to circumvent both under-transfer and negative-transfer.

3 Deep Relationship Networks

In this work, we learn multiple tasks by jointly modeling transferable features and task relationships. Given T supervised learning tasks with data $\{\mathbf{X}_t, \mathbf{y}_t\}_{t=1}^T$, where $\mathbf{X}_t \in \mathbb{R}^{D_0 \times N_t}$ and $\mathbf{y}_t \in \mathbb{R}^{N_t}$ are the design matrix and label vector of the t -th task, respectively drawn from D_0 -dimensional feature space and C -cardinality label space, i.e. each training example $\mathbf{x}_n^t \in \mathbb{R}^{D_0}$ and $y_n^t \in \{1, \dots, C\}$. Our goal is to construct a deep neural network for multiple tasks $y_n^t = f_t(\mathbf{x}_n^t)$ that is able to learn transferable features and adaptive task relationships to bridge different tasks effectively and robustly.

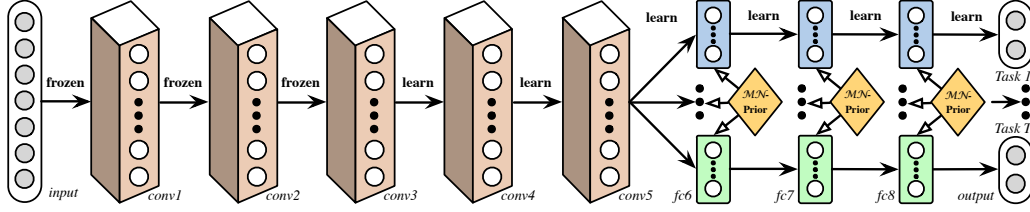


Figure 1: The proposed DRN architecture for learning multiple tasks. Since deep features eventually transition from general to specific along the network [19]: (1) convolutional layers $conv1$ – $conv5$ can learn transferable features, hence their parameters $\{\mathcal{F}^\ell\}_{\ell=1}^5$ are shared among the multiple tasks; (2) fully connected layers $fc6$ – $fc8$ are tailored to fit task-specific variations, hence their task-specific parameters $\{\mathcal{F}_t^\ell\}_{\ell=6}^8$ are jointly modeled via matrix normal priors for learning the task relationships.

3.1 Model

We start with the deep convolutional neural network (CNN) [23], a strong model to learn transferable features that are well adaptive to multiple tasks [18, 19, 22]. The main challenge is that in multi-task learning, each task is provided with only a limited amount of labeled data, which is insufficient to build reliable classifiers without overfitting. In this sense, it is vital to model the task relationships through which each pair of tasks can help with each other if they are related in the parameter space, and can remain independent if they are unrelated to mitigate negative-transfer. With this principle, we design a Deep Relationship Network (DRN) that can exploit both feature transferability and task relationship to enable multi-task learning. Figure 1 gives an illustration of the proposed DRN model.

We extend the AlexNet architecture [23], which is comprised of five convolutional layers ($conv1$ – $conv5$) and three fully connected layers ($fc6$ – $fc8$). Each fc layer ℓ learns a nonlinear mapping $\mathbf{h}_n^\ell = a^\ell(\mathbf{W}^\ell \mathbf{h}_n^{\ell-1} + \mathbf{b}^\ell)$, where \mathbf{h}_n^ℓ is the ℓ -th layer hidden representation of point \mathbf{x}_n , \mathbf{W}^ℓ and \mathbf{b}^ℓ are the weight and bias parameters of the ℓ -th layer, and a^ℓ is the activation function, taken as rectifier units (ReLU) $a^\ell(\mathbf{x}) = \max(\mathbf{0}, \mathbf{x})$ for hidden layers or softmax units $a^\ell(\mathbf{x}) = e^{\mathbf{x}} / \sum_{j=1}^{|\mathbf{x}|} e^{x_j}$ for the output layer. Denote by $\mathcal{F}_t^\ell = \{\mathbf{W}_t^\ell, \mathbf{b}_t^\ell\}$ the network parameters of the t -th task in the ℓ -th layer, and $\mathcal{F}_t = \{\mathcal{F}_t^\ell\}_{\ell=1}^8$ the set of network parameters for the t -th task. The empirical error of CNN is

$$\min_{\mathcal{F}_t \in \mathcal{F}_t} \sum_{n=1}^{N_t} J(f_t(\mathbf{x}_n^t), y_n^t), \quad (1)$$

where J is the cross-entropy loss function, and $f_t(\mathbf{x}_n^t)$ is the conditional probability that the CNN assigns \mathbf{x}_n^t to label y_n^t . We will not describe how to compute the convolutional layers as these layers can learn transferable features [19] and we will simply share their parameters $\{\mathcal{F}_t^\ell = \mathcal{F}^\ell\}_{\ell=1}^5$ across different tasks, without modeling the task relationships in these layers. To benefit from pre-training and fine-tuning, we copy these layers from a model pre-trained from ImageNet 2012 [19, 28], freeze $conv1$ – $conv3$ and fine-tune $conv4$ – $conv5$, which can preserve the efficacy of fragile co-adaptation.

As revealed by the latest literature findings [19], the deep features in standard CNNs must eventually transition from general to specific along the network, and the feature transferability decreases while the task discrepancy increases, making the features in higher layers $fc6$ – $fc8$ unsafely transferable across different tasks. In other words, the fc layers are tailored to their original task at the expense of degraded performance on the target task, which may deteriorate multi-task learning based on deep neural networks. Most previous methods generally assume that the multiple tasks are well correlated given the shared representation learned by the feature layers $conv1$ – $fc7$ of the deep neural network [15, 16, 17, 18]. However, it may be vulnerable if different tasks are not well correlated in the deep features, which is common as higher layers are not safely transferable and tasks may be dissimilar. Another issue not well studied is how to explore multi-class task relationships in multi-task learning.

In this work, we model feature transferability and task relationship based on multiple task-specific layers and multiple class-shared covariances in a Bayesian framework. Denote by $\{\mathbf{X}_t\} = \{\mathbf{X}_t\}_{t=1}^T$, $\{\mathbf{y}_t\} = \{\mathbf{y}_t\}_{t=1}^T$ the data of T tasks, by $\mathbf{w}_{t,c}^\ell$ the network parameter associated with the c -th unit (c -th class for output layer) of the t -th task in the ℓ -th layer, by $\{\mathbf{W}_t^\ell\} = \{\mathbf{W}_t^\ell : 1 \leq t \leq T, 1 \leq \ell \leq L\}$

the set of network parameters for all T tasks, where $\mathbf{W}_t^\ell \triangleq [\mathbf{w}_{t,1}^\ell \dots \mathbf{w}_{t,C_\ell}^\ell] \in \mathbb{R}^{D_\ell \times C_\ell}$ is the network parameter of the t -th task in the ℓ -th layer, D_ℓ and C_ℓ are the number of units in layers $\ell - 1$ and ℓ , respectively. Note that, D_0 is the input dimension and $C_8 = C$ is the label set cardinality. The Maximum a Posteriori (MAP) estimation of the model parameters given data is formalized as

$$\begin{aligned} p\left(\{\mathbf{W}_t^\ell\}_t^\ell \mid \{\mathbf{X}_t\}, \{\mathbf{y}_t\}\right) &= p\left(\{\mathbf{W}_c^\ell\}_c^\ell\right) \times p\left(\{\mathbf{y}_t\} \mid \{\mathbf{X}_t\}, \{\mathbf{W}_c^\ell\}_c^\ell\right) \\ &= \prod_{\ell=L_0}^L \prod_{c=1}^{C_\ell} p\left(\mathbf{W}_c^\ell\right) \times \prod_{t=1}^T \prod_{n=1}^{N_t} p\left(y_n^t \mid \mathbf{x}_n^t, \{\mathbf{W}_t^\ell\}_t^\ell\right), \end{aligned} \quad (2)$$

where $\mathbf{W}_c^\ell \triangleq [\mathbf{w}_{1,c}^\ell \dots \mathbf{w}_{T,c}^\ell] \in \mathbb{R}^{D_\ell \times T}$ is the network parameter of all T tasks associated with the c -th unit in the ℓ -th layer (again, will be the c -th category if it is the output layer), and note that $\{\mathbf{W}_t^\ell\}_t^\ell = \{\mathbf{W}_c^\ell\}_c^\ell = \{\mathbf{W}_c^\ell : 1 \leq c \leq C_\ell, 1 \leq \ell \leq L\}$. Here we adopt the category-wise network parameter \mathbf{W}_c^ℓ instead of the task-wise network parameter \mathbf{W}_t^ℓ because it is technically more viable to model the task relationship with \mathbf{W}_c^ℓ as it contains the category-wise parameters for all T tasks. An important assumption of multi-class classification is that different categories are independent, which is why multi-class problems can be broken into one-vs-rest binary-class problems. We further assume that for the parameter prior $p\left(\{\mathbf{W}_c^\ell\}_c^\ell\right)$, the network parameter of each layer is independent on the network parameters of the other layers, which is a common assumption made by most neural network methods [24]. Finally, we assume when the parameter is sampled from the prior, all tasks are independent. These three assumptions lead to the factorization of the posteriori in Equation (2).

The maximum likelihood estimation (MLE) part in Equation (2) will be modeled by the deep CNN in Equation (1), which can learn transferable features in its lower-layers (*conv1-conv5*) to enhance multi-task learning. Using this property, we opt to share the network parameters of all these layers by $\mathbf{W}_t^\ell = \mathbf{W}^\ell, 1 \leq \ell < L_0$, where $L_0 = 6$, and a different value of L_0 is also possible, depending on the sample sizes and the number of task parameters. This parameter sharing strategy is a relaxation of existing methods [16, 17, 18], which share all lower-layers except for the classifier layer. We do not share task-specific layers (*fc6-fc8*) so as to potentially mitigate the negative-transfer [19, 20].

The prior part in Equation (2) is crucial as it should be able to model the task relationship adaptively. In this paper, we define the following prior based on Gaussian and matrix normal distributions [21]:

$$p\left(\mathbf{W}_c^\ell\right) = \left(\prod_{t=1}^T \mathcal{N}_{D_\ell}\left(\mathbf{w}_{t,c}^\ell \mid \mathbf{m}_c^\ell, \epsilon^2 \mathbf{I}_{D_\ell}\right)\right) \mathcal{MN}_{D_\ell \times T}\left(\mathbf{W}_c^\ell \mid \mathbf{0}_{D_\ell \times T}, \mathbf{I}_{D_\ell}, \mathbf{\Omega}^\ell\right), \quad (3)$$

where $\mathbf{m}_c^\ell \in \mathbb{R}^{D_\ell}$ is the mean parameter of the isotropic Gaussian distribution, and $\mathbf{\Omega}^\ell$ is the covariance parameter of the matrix normal distribution, which is defined as $\mathcal{MN}_{D \times T}(\mathbf{W} \mid \mathbf{M}, \mathbf{\Sigma}, \mathbf{\Omega}) = \frac{\exp\left(-\frac{1}{2} \text{tr}\left(\mathbf{\Sigma}^{-1}(\mathbf{W}-\mathbf{M})\mathbf{\Omega}^{-1}(\mathbf{W}-\mathbf{M})^\top\right)\right)}{(2\pi)^{DT/2} |\mathbf{\Sigma}|^{T/2} |\mathbf{\Omega}|^{D/2}}$. The first term of the prior on \mathbf{W}_c^ℓ is to penalize the complexity of each centered column $\mathbf{w}_{t,c}^\ell - \mathbf{m}_c^\ell$ separately, and the second term is to model the structure of \mathbf{W}_c^ℓ . Specifically, in the Gaussian prior, the mean vector \mathbf{m}_c^ℓ models the common component shared by multiple tasks; and in the matrix normal prior, the row covariance matrix \mathbf{I}_{D_ℓ} models the relationships between features, and the column covariance matrix $\mathbf{\Omega}^\ell$ models the relationships between task parameters $\{\mathbf{w}_{t,c}^\ell\}_{t=1}^T$. Both parameters are learned from data to build adaptive task relationships.

Integrating Equations (1) and (3) and taking the negative logarithm of Equation (2), we obtain the MAP estimation of $\{\mathbf{W}_t^\ell\}_t^\ell$ and the MLE estimation of $\{\mathbf{b}_t^\ell\}_t^\ell$ by solving the optimization problem:

$$\min_{f \in \mathcal{F}, \mathbf{\Omega} \in \mathcal{C}} \sum_{t=1}^T \sum_{n=1}^{N_t} J\left(f_t\left(\mathbf{x}_n^t\right), y_n^t\right) + \sum_{\ell=L_0}^L \sum_{c=1}^{C_\ell} \text{tr}\left(\mathbf{W}_c^\ell\left(\bar{\mathbf{\Omega}}^\ell + \lambda \mathbf{L}\right) \mathbf{W}_c^{\ell \top}\right), \quad (4)$$

where the penalty parameter $\lambda = 1/\epsilon^2$, $\bar{\mathbf{\Omega}}^\ell \triangleq \left(\mathbf{\Omega}^\ell\right)^{-1}$ is the inverse covariance matrix for modeling the task relationship, and \mathbf{L} is the graph Laplacian matrix for modeling the common task component, with $L_{tt'} = 1 - 1/T$ if $t = t'$ and $L_{tt'} = -1/T$ otherwise. Note that we should jointly minimize the concave complexity penalty for $\mathbf{\Omega}^\ell$, i.e. $\sum_{\ell=L_0}^L \ln |\mathbf{\Omega}^\ell|$. We take similar strategy as [11, 13] and minimize its convex upper bound $\sum_{\ell=L_0}^L \left(\text{tr}\left(\mathbf{\Omega}^\ell\right) - T\right)$, which is reduced to the convex constraint:

$$\mathcal{C} \triangleq \left\{\mathbf{\Omega}^\ell : \mathbf{\Omega}^\ell \in \mathbb{R}^{T \times T}, \mathbf{\Omega}^\ell \succeq 0, \text{tr}\left(\mathbf{\Omega}^\ell\right) = 1, L_0 \leq \ell \leq L\right\}. \quad (5)$$

Our work contrasts from previous multi-task deep learning [15, 16, 17, 18] and relationship learning methods [11, 12, 13] in several important aspects: (1) We define the prior on multiple task-specific layers $L_0 \leq \ell \leq L$, while previous deep learning methods do not define the prior and merely rely on the bet that the shared deep features are good enough for multi-task learning (which may not be true due to Yosinski et al. [19]), and previous relationship learning methods are not designed in a deep architecture (which cannot learn transferable features for multi-task learning); (2) We define the prior by a task covariance matrix Ω^ℓ shared by all categories (which can learn the task relationship more accurately when category-wise data is scarce), while previous relationship learning methods define the prior for binary or regression problems; (3) We define the prior with a task mean parameter \mathbf{m}_c^ℓ that captures the common task component of multiple tasks, while previous relationship learning methods define the prior with zero-mean Gaussian that cannot model the common task component. Hence the proposed DRN model is more effective for multi-task learning and robust to outlier tasks.

3.2 Algorithm

As the optimization problem (4) is jointly non-convex with respect to the network parameters $\{\mathbf{W}_t^\ell\}_t^\ell$ and task covariances $\{\Omega^\ell\}^\ell$, we adopt an alternating method that optimizes one set of variables with the others fixed. We first update \mathbf{W}_t^ℓ , which needs reformulation of objective (4) in terms of $\{\mathbf{W}_t^\ell\}_t^\ell$:

$$O = \sum_{t=1}^T \sum_{n=1}^{N_t} J(f_t(\mathbf{x}_n^t), y_n^t) + \sum_{\ell=L_0}^L \sum_{t=1}^T \sum_{t'=1}^T (\bar{\Omega}_{tt'}^\ell + \lambda L_{tt'}) \text{tr}(\mathbf{W}_t^\ell \mathbf{W}_{t'}^{\ell \top}). \quad (6)$$

When we train deep CNN by mini-batch stochastic gradient descent (SGD), we only need to consider the gradient of objective (6) corresponding to each data point (\mathbf{x}_n^t, y_n^t) , which can be computed as

$$\frac{\partial O(\mathbf{x}_n^t, y_n^t)}{\partial \mathbf{W}_t^\ell} = \frac{\partial J(f_t(\mathbf{x}_n^t), y_n^t)}{\partial \mathbf{W}_t^\ell} + 2 \sum_{t'=1}^T (\bar{\Omega}_{tt'}^\ell + \lambda L_{tt'}) \mathbf{W}_{t'}^\ell. \quad (7)$$

Such a mini-batch SGD can be easily implemented via the Caffe framework for CNNs [28]. Since training a deep CNN requires a large amount of labeled data, which is prohibitive for many multi-task learning problems, we fine-tune from an AlexNet model pre-trained on ImageNet 2012 as [19]. In each epoch of SGD, with the updated $\{\mathbf{W}_t^\ell\}_t^\ell$, we can update $\{\Omega^\ell\}^\ell$ by a close-form solution as

$$\Omega^\ell = \frac{(\mathbf{A}^\ell)^{1/2}}{\text{tr}((\mathbf{A}^\ell)^{1/2})}, \text{ where } A_{tt'}^\ell = \text{tr}(\mathbf{W}_t^\ell \mathbf{W}_{t'}^{\ell \top}), \text{ and } \bar{\Omega}^\ell \triangleq (\Omega^\ell)^{-1}. \quad (8)$$

Though the derivation is similar to [11], our method learns the task relationship Ω^ℓ from *multiple* classes, while [11] learns the task relationship Ω^ℓ from *binary* classes. When the labeled data is very limited for each category, the binary-class method [11] may not infer the task relationship accurately.

The DRN algorithm scales linearly to the sample size. For each iteration, the time cost of all convolutional layers is $O(\sum_{\ell=1}^{L_{cv}} N F_{\ell-1} S_\ell^2 \cdot F_\ell M_\ell^2)$, where L_{cv} is the number of convolutional layers, F_ℓ is the number of filters in the ℓ -th layer, S_ℓ is the spatial size of the filter, and M_ℓ is the size of output feature map; and the time cost of all fully connected layers is $O(\sum_{\ell=L_{cv}+1}^L N(D_\ell^2 C_\ell + T D_\ell C_\ell))$. Finally, the time complexity for updating task relationship matrices is $O(\sum_{\ell=L_0}^L (T^2 D_\ell^2 C_\ell + T^3))$.

3.3 Discussion

We consider a fine-grained task relationship where different tasks may be correlated in different ways for different categories. This may be beneficial when the training set for each class is relatively large so that class-wise task relationship can be learned accurately, and the class-conditional distributions are very different across tasks. This leads to a DRN variant for learning class-wise task relationships:

$$\min_{f \in \mathcal{F}, \Omega \in \mathcal{C}} \sum_{t=1}^T \sum_{n=1}^{N_t} J(f_t(\mathbf{x}_n^t), y_n^t) + \sum_{\ell=L_0}^L \sum_{c=1}^{C_\ell} \text{tr}(\mathbf{W}_c^\ell (\bar{\Omega}_c^\ell + \lambda \mathbf{L}) \mathbf{W}_c^{\ell \top}), \quad (9)$$

where Ω_c^ℓ is the class-wise task relationship, which decouples Equation (9) for different categories, i.e. equivalent to C one-vs-rest binary problems, but training C deep networks is more demanding.

Learning shared features by exploiting feature covariance has been extensively studied for multi-task feature learning [2, 3, 5, 4, 6]. We generalize DRN to handle the case that both the tasks and features are correlated by modifying the matrix normal prior as $\mathcal{MN}_{D_\ell \times T}(\mathbf{W}_c^\ell | \mathbf{0}_{D_\ell \times T}, \Sigma^\ell, \Omega^\ell)$ in Equation (3), which leads to a new DRN variant for learning both feature and task relationships:

$$\min_{f, \Sigma, \Omega} \sum_{t=1}^T \sum_{n=1}^{N_t} J(f_t(\mathbf{x}_n^t), y_n^t) + \sum_{\ell=L_0}^L \sum_{c=1}^{C_\ell} \text{tr} \left(\bar{\Sigma}^\ell \mathbf{W}_c^\ell (\bar{\Omega}^\ell + \lambda \mathbf{L}) \mathbf{W}_c^{\ell T} \right), \quad (10)$$

where Σ^ℓ is the feature covariance matrix for encoding the feature relationship and $\bar{\Sigma}^\ell$ is its inverse. Note that deep networks can implicitly learn from the feature covariance via the network parameters.

4 Experiments

We compare the DRN model with state-of-the-art multi-task learning methods on real-world object recognition datasets, and verify the efficacy of the multi-layer and multi-class relationship learning. Different from most methods that use multi-task datasets where each task is binary classification or univariate regression, we evaluate on multi-task datasets where each task is multi-class classification.

4.1 Setup

Office-Caltech [29, 30] This dataset is the standard benchmark for multi-task learning and transfer learning. The Office part [29] consists of 4,652 images in 31 categories collected from three distinct domains (tasks): *Amazon* (**A**), which contains images downloaded from `amazon.com`, *Webcam* (**W**) and *DSLR* (**D**), which are images taken by Web camera and digital SLR camera under different environmental variations. This dataset is organized by selecting the 10 common categories shared by the Office dataset and the Caltech-256 (**C**) dataset [30], hence it consists of four domains (tasks).

ImageCLEF-DA¹ This dataset is the benchmark for ImageCLEF domain adaptation challenge. It is organized by selecting the 12 common categories shared by the following four public datasets (tasks): Caltech-256 (**C**), ImageNet ILSVRC 2012 (**I**), Pascal VOC 2012 (**P**), and Bing (**B**). The 12 common categories are: aeroplane, bike, bird, boat, bottle, bus, car, dog, horse, monitor, motorbike, and people. For space limitation, we omit the dataset details that can be parsed from the Web. Both datasets are evaluated via SURF features for shallow methods and original images for deep methods.

We compare with a variety of methods: Single-Task Softmax Regression (**STSR**), Multi-Task Feature Learning (**MTFL**) [3], Robust Multi-Task Learning (**RMTL**) [5], Multi-Task Relationship Learning (**MTRL**) [11, 13], and Multi-Task Deep Convolutional Neural Network (**MTCNN**) [18]. Specifically, LR is performed on each task separately, without modeling the shared structure across tasks. MTFL learns the covariance structure of features and different tasks are independent given the covariance structure. RMTL is an extension to MTFL that captures the task relationships using a low-rank structure, and simultaneously identifies the outlier tasks using a group-sparse structure. MTRL infers the task relationships by the task covariance matrix, hence it can be adaptive to the transferability between tasks and circumvent the negative-transfer problem. The original MTCNN is applied to heterogeneous tasks (e.g. face landmark detection and head pose estimation) by sharing all the feature layers and learning multiple task classifiers, while we apply it to homogeneous tasks.

To study the efficacy of jointly learning transferable features and task relationships, we evaluate several variants of DRN: (1) DRN using only one network layer, i.e. layer `fc8` for relationship learning, termed **DRN₈**; (2) DRN using one-vs-rest binary classifier for relationship learning, termed **DRN_{bi}**; (3) DRN without enforcing the task mean in isotropic Gaussian prior, termed **DRN_m**. While using binary classifier for multi-class problems is less effective due to unbalanced classes, and inefficient for large number of categories, most existing methods [3, 5, 11] generally follow this paradigm. The plausible reason is that it may be not easy to extend these methods to natural multi-class formulation.

We mainly follow standard evaluation protocol for multi-task learning and randomly select 5%, 10%, and 20% samples from each task as the training set and utilize the rest of the samples as the test set [11, 13, 5]. Note that the sample sizes may be imbalanced across different tasks, e.g. **C** and **A** are large domains with thousands of examples while **W** and **D** are small domains with only hundreds of

¹<http://imageclef.org/2014/adaptation>

Table 1: Multi-class accuracy on the *Office-Caltech* dataset with standard evaluation protocol [11].

Method	5%					10%					20%				
	A	W	D	C	Avg	A	W	D	C	Avg	A	W	D	C	Avg
STSR	45.0	50.8	39.2	32.1	41.8	51.9	65.0	46.6	35.4	49.7	59.6	70.4	57.8	39.9	56.0
MTFL	45.5	45.6	43.0	34.0	42.0	51.2	64.3	47.0	33.9	49.1	60.4	73.7	64.3	39.1	59.4
RMTL	47.0	43.9	43.9	27.5	40.6	50.9	58.5	48.3	32.9	47.6	55.0	74.3	59.5	39.8	57.1
MTRL	43.2	50.4	41.2	28.3	40.8	51.8	67.2	51.1	33.8	50.9	61.3	76.8	67.7	39.0	59.9
MTCNN	74.2	87.1	82.9	75.1	79.8	89.0	91.9	87.4	82.8	87.8	92.6	95.4	90.0	86.9	91.2
DRN ₈	92.2	96.9	97.4	87.0	93.4	93.2	97.6	97.8	87.1	93.9	93.7	97.4	99.9	88.3	94.9
DRN _{bi}	91.6	96.8	97.9	86.6	93.2	93.0	98.2	97.1	87.2	93.9	92.4	97.1	96.4	90.5	94.1
DRN _m	92.4	96.1	98.6	87.3	93.6	93.7	97.9	97.9	86.5	94.0	93.5	97.9	98.7	89.8	95.0
DRN	92.5	97.5	97.9	87.5	93.8	93.6	98.6	98.6	87.3	94.5	94.4	98.3	99.9	89.1	95.5

Table 2: Multi-class accuracy on the *ImageCLEF-DA* dataset with standard evaluation protocol [11].

Method	5%					10%					20%				
	C	I	P	B	Avg	C	I	P	B	Avg	C	I	P	B	Avg
STSR	48.6	28.8	20.8	20.6	29.7	54.1	31.2	21.5	23.1	32.5	62.2	36.5	25.4	25.1	37.3
MTFL	44.6	27.3	19.6	20.0	27.9	53.2	30.4	23.0	22.5	32.3	62.5	37.0	24.7	24.6	37.2
RMTL	44.2	27.5	23.8	21.4	29.2	54.2	31.3	23.7	23.0	33.1	60.8	36.3	24.7	27.4	37.3
MTRL	51.6	31.0	23.9	17.0	30.9	61.3	31.9	25.2	28.9	36.8	63.8	41.0	28.0	31.5	41.0
MTCNN	84.9	67.0	55.1	31.1	59.5	89.1	76.1	55.7	45.0	66.5	91.7	80.0	60.2	51.1	70.7
DRN ₈	87.0	74.4	59.8	45.6	66.7	89.1	82.2	60.4	49.3	70.2	91.1	84.1	65.7	54.1	73.7
DRN _{bi}	89.1	76.7	60.5	45.1	67.8	88.9	80.9	61.5	50.7	70.5	91.1	83.5	65.7	55.7	74.0
DRN _m	88.4	75.8	60.2	38.6	65.7	89.3	80.7	60.7	49.4	70.0	91.7	83.5	61.1	52.8	72.3
DRN	89.1	77.9	61.9	47.0	69.0	88.7	81.9	62.3	51.3	71.0	91.3	84.4	66.3	53.3	73.8

examples. In this regard, the evaluation protocol can naturally reflect the performance of multi-task learning with different sample sizes. We compare the averages of accuracy for all tasks based on 10 random experiments, while standard errors are insignificant and are not listed. We perform parameter selection for all methods using five-fold cross-validation on the training set, which is consistent with the baseline methods [3, 5, 11]. For CNN-based methods, we adopt the fine-tuning architecture [19], however, due to limited training examples in our datasets, we fix convolutional layers *conv1–conv3* that were copied from pre-trained model, fine-tune *conv4–conv5* and fully connected layers *fc6–fc7*, and train classifier layer *fc8*, both via back propagation. As the classifier is trained from scratch, we set its learning rate to be 10 times that of the lower layers. We use stochastic gradient descent (SGD) with 0.9 momentum and the learning rate annealing strategy implemented in Caffe [28], and cross-validate the learning rate between 10^{-5} and 10^{-2} by a multiplicative step-size $10^{0.5}$.

4.2 Results and Discussion

The multi-task classification results on the multi-class datasets Office-Caltech and ImageCLEF-DA based on 5%, 10%, and 20% sampled training data are shown in Tables 1 and 2, respectively. We can observe that the proposed DRN model significantly outperforms the comparison methods on all multi-task problems. The substantial performance boost verifies that our deep relationship networks via multi-layer and multi-class relationship learning is able to learn both safely transferable features and adaptive task relationships, which establishes more effective and robust multi-task deep learning.

From the experimental results, we can make the following observations. (1) Conventional shallow multi-task learning methods MTFL, RMTL, and MTRL generally outperform single-task learning method STSR, which confirms the motivation of jointly learning multiple tasks by exploiting shared structures. Among the shallow multi-task methods, MTRL gives the best performance, showing that exploiting task relationship is more effective than extracting shallow feature subspace for multi-task learning. (2) Current state-of-the-art multi-task deep learning method MTCNN further outperforms conventional shallow multi-task learning methods by a very large margin, which certifies the vital importance of learning deep transferable features to enable knowledge transfer across different tasks. However, MTCNN assumes shared network parameters for all feature layers (*conv1–fc7*) and learns

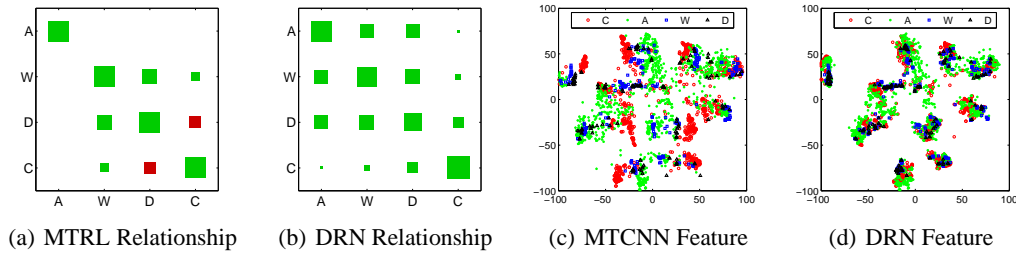


Figure 2: Hinton diagram of task relationship (a)(b) and t-SNE embedding of deep features (c)(d).

the classifier layer $fc8$ independently without inducing the task relationship. This results in negative-transfer in the feature layers [19] and under-transfer in the classifier layer. A defense for MTCNN is that it is designed for heterogeneous multi-task learning where task relationship is difficult to model.

To dive deeper into DRN, we present the results of three variants of DRN: DRN_8 , DRN_{bi} and DRN_m , all significantly outperform MTCNN but generally underperform DRN, which verify our motivation that jointly learning transferable features and adaptive task relationships can bridge multiple tasks more effectively. (1) The disadvantage of DRN_8 is that it does not learn the adaptive task relationship in the lower fully connected layers $fc6$ – $fc7$, which are not completely transferable and may result in negative-transfer [19]. (2) The shortcoming of DRN_{bi} is that it does not learn the task relationship based on multiple categories, hence the inferred class-wise task relationship may be inaccurate when labeled data is very limited, however, its performance will catch up with DRN when the class-wise training data grows large. (3) The weakness of DRN_m is that it does not capture the common task component, hence it may result in under-transfer if the common task component really exists, especially for the lower feature layers of the deep network that are more safely transferable [19]. The proposed DRN model takes full advantages of all these factors and establishes the best performance.

4.3 Visualization Analysis

We first show that DRN can better learn adaptive task relationships with deep features than MTRL with shallow features, by visualizing the task covariance matrices Ω learned by MTRL and DRN in Figures 2(a) and 2(b), respectively. Prior knowledge on task similarity in the Office-Caltech dataset [29] describes that tasks **A**, **W** and **D** are more similar with each other while they are significantly dissimilar to task **C**. DRN successfully captures this prior task relationship and further enhances task correlation across dissimilar tasks, which establishes stronger transferability for multi-task learning. Furthermore, all tasks are positively correlated (green color) in DRN, implying that all tasks can better reinforce each other. However, some of the tasks (**D** and **C**) are still negatively correlated (red color) in MTRL, implying these tasks should be drawn far apart and cannot improve with each other.

To demonstrate the transferability of DRN features, we follow [28, 20] and visualize in Figures 2(c) and 2(d) the t-SNE embeddings of the images in the Office-Caltech dataset with MTCNN features and DRN features, respectively. We observe that compared with MTCNN features, the data points with DRN features are discriminated better across different categories, i.e. each category has small intra-class variance and large inter-class margin; and the data points are also aligned better across different tasks, i.e. the embeddings of different tasks overlap well, which implies that different tasks can reinforce each other effectively and improve category discrimination performance. This verifies that with joint relationship discovery, DRN learns more transferable features for multi-task learning.

5 Conclusion

We proposed a deep relationship network (DRN) model that integrates standard neural networks with matrix normal priors over the network parameters of all task-specific layers. The priors define the covariance structure over tasks and enable inductive transfer across related tasks. We devised a learning algorithm that fine-tunes from a pre-trained deep network and learns transferable features and task relationships jointly. Experiments show that the model achieves superior results on standard object recognition datasets. Future work includes data-dependent priors for multi-task deep learning.

References

- [1] R. Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.
- [2] R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.
- [3] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.
- [4] J. Chen, L. Tang, J. Liu, and J. Ye. A convex formulation for learning a shared predictive structure from multiple tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(5):1025–1038, 2013.
- [5] J. Chen, J. Zhou, and J. Ye. Integrating low-rank and group-sparse structures for robust multi-task learning. In *KDD*, 2011.
- [6] D. Hernández-Lobato, J. M. Hernández-Lobato, and Z. Ghahramani. A probabilistic model for dirty multi-task feature selection. In *ICML*, 2015.
- [7] L. Jacob, J.-P. Vert, and F. R. Bach. Clustered multi-task learning: A convex formulation. In *NIPS*, 2009.
- [8] Z. Kang, K. Grauman, and F. Sha. Learning with whom to share in multi-task feature learning. In *ICML*, 2011.
- [9] A. Kumar and H. Daume III. Learning task grouping and overlap in multi-task learning. *ICML*, 2012.
- [10] T. Evgeniou and M. Pontil. Regularized multi-task learning. In *KDD*, 2004.
- [11] Y. Zhang and D.-Y. Yeung. A convex formulation for learning task relationships in multi-task learning. In *UAI*, 2010.
- [12] Y. Zhang and J. Schneider. Learning multiple tasks with a sparse matrix-normal penalty. In *NIPS*, 2010.
- [13] Y. Zhang and D.-Y. Yeung. A regularization approach to learning task relationships in multitask learning. *ACM Transactions on Knowledge Discovery from Data*, 8(3):12, 2014.
- [14] C. Ciliberto, Y. Mroueh, T. Poggio, and L. Rosasco. Convex learning of multiple tasks and their structure. In *ICML*, 2015.
- [15] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.
- [16] N. Srivastava and R. Salakhutdinov. Discriminative transfer learning with tree-based priors. In *NIPS*, 2013.
- [17] W. Ouyang, X. Chu, and X. Wang. Multisource deep learning for human pose estimation. In *CVPR*, 2014.
- [18] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Facial landmark detection by deep multi-task learning. In *ECCV*, 2014.
- [19] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *NIPS*, 2014.
- [20] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015.
- [21] A. K. Gupta and D. K. Nagar. *Matrix variate distributions*. Chapman & Hall, 2000.
- [22] J. Hoffman, S. Guadarrama, E. Tzeng, R. Hu, J. Donahue, R. Girshick, T. Darrell, and K. Saenko. LSDA: Large scale detection through adaptation. In *NIPS*, 2014.
- [23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.
- [24] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.
- [25] X. Glorot, A. Bordes, and Y. Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *ICML*, 2011.
- [26] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *ICML*, 2011.
- [27] G. Heigold, V. Vanhoucke, A. Senior, P. Nguyen, M. Ranzato, M. Devin, and J. Dean. Multilingual acoustic models using distributed deep neural networks. In *ICASSP*, 2013.
- [28] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM Multimedia*, 2014.
- [29] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*, 2010.
- [30] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2012.