Task Relation Networks

Jianshu Li [1,2] Pan Zhou [1] Yunpeng Chen [1] Jian Zhao [1]
Sujoy Roy [2] Shuicheng Yan [1] Jiashi Feng [1] Terence Sim [1]

[1]National University of Singapore [2]SAP Machine Learning Singapore

{jianshu, pzhou, chenyunpeng, zhaojian90}@u.nus.edu sujoy.roy@sap.com
{elefjia, eleyans}@nus.edu.sg tsim@comp.nus.edu.sg

Abstract

Multi-task learning is popular in machine learning and computer vision. In multitask learning, properly modeling task relations is important for boosting the performance of jointly learned tasks. Task covariance modeling has been successfully used to model the relations of tasks but is limited to homogeneous multi-task learning. In this paper, we propose a feature based task relation modeling approach, suitable for both homogeneous and heterogeneous multi-task learning. First, we propose a new metric to quantify the relations between tasks. Based on the quantitative metric, we then develop the task relation layer, which can be combined with any deep learning architecture to form task relation networks to fully exploit the relations of different tasks in an online fashion. Benefiting from the task relation layer, the task relation networks can better leverage the mutual information from the data. We demonstrate our proposed task relation networks are effective in improving the performance in both homogeneous and heterogeneous multi-task learning settings through extensive experiments on computer vision tasks.

1. Introduction

Multi-task learning has been the focus in the community of machine learning and computer vision. Learning with closely related tasks is believed to be helpful for boosting the performance of the involved tasks, which has been verified by considerable research in many areas such as computer vision [25, 16], and other areas like natural language processing [7, 23]. To better understand multi-task learning, a lot of research effort has been made to study better ways of modeling and leveraging the relations among different tasks.

Existing methods for task relation modeling in multi-task learning can be categorized into task covariance based and feature based ones. For task covariance based modeling, traditional methods usually build a task covariance matrix between the learned parameters of different tasks [2, 24] to characterize the task relations. Although effective, task covariance is only applicable to homogeneous multi-task learning setting, where the output spaces of different tasks have the same dimension. Essentially in homogeneous multi-task learning, the special setting on identical output dimensions determines that different linear models for different tasks have the same structure. This makes the relation modeling easier, and a task covariance matrix can be obtained conveniently from the parameters for different tasks. Very recently, [15] extends such classic task covariance modeling from shallow linear models to deep models by separately examining the parameters per layer. In their method, the task covariance matrix is learned together with the whole model. Such explicit modeling of task relations is shown to be effective in multi-task learning setting for capturing mutual information among different tasks and improving their performance.

For heterogeneous multi-task learning, where different tasks have different output dimensions, the task covariance relation modeling is no longer valid as parameters for different models have different structures. This problem is only partially addressed in [15, 22] because these models are only able to model the relations between hidden layer parameters, which are of the same shape and cannot model task relations at the critical output layer. Such a kind of modeling is arguably not optimal and may harm the multi-task model capability and final performance. Essentially, it is not clear how to fully model task relations in heterogeneous multi-task learning using the task covariance.

Apart from task relation modeling using task covariance, the feature based task relation modeling is also widely used in the literature. Based on the similarity of feature vectors, multiple tasks can be grouped together [12] to facilitate the learning process. [1] defines that two tasks are similar if

they use the same feature to make predictions [20]. [5] uses the distance between the response maps to decide if two tasks are similar tasks. [16] uses cross-stitch networks to learn the sharing structure of common and task-specific features for two tasks. It allows different tasks to share intermediate features with each other, and the sharing degree can roughly reflect if the involved tasks are closely or loosely related. However, none of the works in the feature-based task relation modeling defines explicit metrics to quantitatively measure the relations between different tasks, making them less principled and unclear on how to further improve.

Considering the demand for explicit metrics for task relations in feature-based relation modeling, and the inherent limitation of task covariance based modeling, we propose a similarity metric to quantitatively measure the relations between tasks in both homogeneous and heterogeneous multi-task learning. The proposed similarity metric, named Statistical Task Relation (STR), measures the relations between tasks from a statistical point of view (details in Sec. 3.1). The metric is developed based on our observation of isotopic multi-task learning, where different outputs are associated with the same input. In such cases, the multi-task models (both homogeneous and heterogeneous) are essentially learned mappings from the same input data to different outputs. Thus what distinguishes one task from another are the features the inputs are mapped to by the model. Therefore we measure the similarity between the features in the output space (*i.e.* the predictions), and use it as the metric for relations between two tasks.

Based on the proposed similarity metric STR, we further develop a Task Relation Layer (TRL), which serves as the workhorse to measure the similarity between the output features *dynamically* in a deep neural network. Different from [15] which models the relation of the model *weights* in each layer, we directly model the relation between the *features*. An online learning algorithm is also proposed such that TRL can be inserted into any deep learning architecture and end-to-end trained. When incorporating the TRL into a deep neural network, the resultant Task Relation Net (TRN) can better leverage the mutual information within the tasks. We demonstrate that TRN can effectively increase the performance of baseline models with the explicit modeling of relations between tasks in both homogeneous and heterogeneous multi-task learning, as shown by our experiments on facial landmark localization and attribute classification tasks.

2. Related Work

2.1. Multi-Task Feature Learning and Relation Learning

Multi-task learning aims to learn to perform multiple tasks by exploiting their shared information for better generalization capability. Traditionally, multi-task learning schemes can be categorized into multi-task feature learning and multi-task relation learning. The first category mainly focuses on learning features shared for different tasks, such that it enables implicit data augmentation and mitigates representation bias [20]. The second category usually uses task covariance [2, 24] to model the relationship between tasks to achieve mutual performance boosts. Different from them, our proposed TRN explicitly models the relations between the features, so that more favorable features can be learned for improving upon multiple tasks.

2.2. Deep Models for Multi-Task Learning

Multi-task learning has been commonly used in deep learning. The structure of one trunk and several branches in the "Share-and-Split" scheme is popular in deep models. While there are quite a number of successful deep models learning different tasks with such a structure (see [18, 21, 17, 13]), very few have studied the relations between different tasks. [16] uses cross-stitch networks to learn the sharing structure of common and task-specific representations for two tasks and [5] presents a similar idea and extends it to more than two tasks. However, explicit modeling of task relations is not present in these works. Different from these works, our proposed TRN explicitly presents a metric to measure the similarity between tasks and uses the similarity to regularize the learning of the network.

2.3. Homogeneous and Heterogeneous Multi-Task Learning

Compared with homogeneous multi-task learning that requires different tasks to have the same output dimension, heterogeneous multi-task learning is attracting more research attention due to its flexibility of modeling various types of tasks. [6] uses a heterogeneous model to predict multiple heterogeneous face attributes. In [25], a heterogeneous model is used to predict facial landmark locations and facial attributes. [10] presents a model for multilingual speech processing. In [22], a tensor factorization approach is used to decompose only the weights in the shared layers in a heterogeneous deep multi-task model, and the weights in the last layer are un-modeled due to the incompatibility of the dimensions of model weights. A similar situation is observed in [15], where the last layer of the heterogeneous multi-task learning model is un-modeled due to the differences in output spaces. So none of the works above fully model the task relations in the heterogeneous setting. In contrast, the proposed TRN can fully model the task relation in all the task-specific layers in both homogeneous and heterogeneous multi-task learning settings.

3. Method

In this section, we first define the proposed similarity metric Statistical Task Relation (STR). Then we introduce the Task Relation Layer (TRL), which incorporates STR into a deep neural network for estimating the similarity of dynamic features, and an online learning algorithm that enables end-to-end training of the deep model with TRL. With the proposed TRL and the online updating algorithm, the resultant Task Relation Network (TRN) can better exploit mutual information and increase the performance of the tasks.

3.1. STR: Proposed Metric for Task Similarity

We first introduce STR, our proposed similarity metric between two tasks from a statistical point of view. In single-task supervised learning, given a set of input samples $\mathcal{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(N)}\}$, where $\mathbf{x}^{(i)} \in \mathbb{R}^d$, and their corresponding labels $\mathcal{Y}_1 = \{\mathbf{y}_1^{(1)}, \mathbf{y}_1^{(2)}, \cdots, \mathbf{y}_1^{(N)}\}$, where $\mathbf{y}_1^{(i)} \in \mathbb{R}^{n_1}$, the single-task model aims to learn a mapping from the input set $\mathcal{X}$ to the label set $\mathcal{Y}_1$, *i.e.* $f_1(\mathbf{x}^{(i)}) \rightarrow \mathbf{y}_1^{(i)}$. In multi-task learning, in addition to one set of labels $\mathcal{Y}_1$, there are usually extra sets (at least one more) of labels. Without loss of generality, we consider the multi-task learning with two different label sets. The other label set is denoted as $\mathcal{Y}_2 = \{\mathbf{y}_2^{(1)}, \mathbf{y}_2^{(2)}, \cdots, \mathbf{y}_2^{(N)}\}, \mathbf{y}_2^{(i)} \in \mathbb{R}^{n_2}$ associated with the same input set $\mathcal{X}$. Then the multi-task learning model aims to learn mappings from the input to both sets of labels, *i.e.* two mappings, $f_1(\mathbf{x}^{(i)}) \rightarrow \mathbf{y}_1^{(i)}$ and $f_2(\mathbf{x}^{(i)}) \rightarrow \mathbf{y}_2^{(i)}$ that are learned jointly. We denote the two tasks involved as $\mathcal{T}_1 \triangleq \{f_1(\mathbf{x}^{(i)}) \rightarrow \mathbf{y}_1^{(i)}, \forall i \in 1, 2, \cdots, N\}$ and $\mathcal{T}_2 \triangleq \{f_2(\mathbf{x}^{(i)}) \rightarrow \mathbf{y}_2^{(i)}, \forall i \in 1, 2, \cdots, N\}$. According to the output dimensions, multi-task learning can be classified into homogeneous multi-task learning if $n_1 = n_2$, and heterogeneous one otherwise, as mentioned in Sec. 1.

We can see that $\mathcal{T}_1$ and $\mathcal{T}_2$ capture and represent the relevant information in input $\mathcal{X}$ w.r.t. the labels $\mathcal{Y}_1$ and $\mathcal{Y}_2$, respectively. Suppose $\mathcal{T}_1$ and $\mathcal{T}_2$ generate output features (*i.e.* the predictions) $\mathbf{f}_1^{(i)}$ and $\mathbf{f}_2^{(i)}$, respectively, when taking $\mathbf{x}^{(i)}$ as input, *i.e.* $\mathbf{f}_1^{(i)} = f_1(\mathbf{x}^{(i)})$ and $\mathbf{f}_2^{(i)} = f_2(\mathbf{x}^{(i)})$. The relation of $\mathcal{T}_1$ and $\mathcal{T}_2$ can be characterized by the similarity between the output feature sets $\mathcal{F}_1 = \{\mathbf{f}_1^{(1)}, \mathbf{f}_1^{(2)}, \cdots, \mathbf{f}_1^{(N)}\}$ and $\mathcal{F}_2 = \{\mathbf{f}_2^{(1)}, \mathbf{f}_2^{(2)}, \cdots, \mathbf{f}_2^{(N)}\}$, considering they are taking in the same inputs. So we propose to adopt a similarity metric between $\mathcal{F}_1$ and $\mathcal{F}_2$, denoted as $\tau(\mathcal{F}_1, \mathcal{F}_2)$, to quantify the relation between the two tasks $\mathcal{T}_1$ and $\mathcal{T}_2$. In particular, we adopt Canonical Component Analysis (CCA) [9] to find the maximal correlations between the two sets of variables $\mathcal{F}_1$ and $\mathcal{F}_2$ as the similarity between the two tasks $\mathcal{T}_1$ and $\mathcal{T}_2$. We adopt CCA as it is commonly used to find what is common in two sets of variables, and it is applicable even if the two sets of variables are of different dimensions.

Under the framework of CCA, we aim to find two auxiliary vectors $\mathbf{a}$ and $\mathbf{b}$ such that the correlation between $\mathbf{a}^T\mathbf{f}_1^{(i)}$ and $\mathbf{b}^T\mathbf{f}_2^{(i)}$, denoted as ρ, is maximized. Then we use the maximal correlation ρ_m as the similarity between $\mathcal{F}_1$ and $\mathcal{F}_2$, *i.e.* $\tau(\mathcal{F}_1, \mathcal{F}_2) \triangleq \rho_m$.

The correlation between $\mathbf{a}^T\mathbf{f}_1^{(i)}$ and $\mathbf{b}^T\mathbf{f}_2^{(i)}$ is defined as

$$\rho = \frac{\mathbf{a}^T\mathbf{\Sigma}_{12}\mathbf{b}}{\sqrt{\mathbf{a}^T\mathbf{\Sigma}_{11}\mathbf{a}}\sqrt{\mathbf{b}^T\mathbf{\Sigma}_{22}\mathbf{b}}}, \tag{1}$$

where $\mathbf{\Sigma}$ denotes the covariance matrix, and the $\mathbf{f}$ in subscript of the covariance matrix is omitted for conciseness. Here we aim to find the maximal ρ w.r.t. $\mathbf{a}$ and $\mathbf{b}$. Substituting $\mathbf{c} = \mathbf{\Sigma}_{11}^{1/2}\mathbf{a}$ and $\mathbf{d} = \mathbf{\Sigma}_{22}^{1/2}\mathbf{b}$, ρ is re-written as

$$\rho = \frac{\mathbf{c}^T\mathbf{\Sigma}_{11}^{-1/2}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1/2}\mathbf{d}}{\sqrt{\mathbf{c}^T\mathbf{c}}\sqrt{\mathbf{d}^T\mathbf{d}}}. \tag{2}$$

With Cauchy–Schwarz inequality in

$$\begin{aligned}&\left(\mathbf{c}^T\Sigma_{11}^{-1/2}\Sigma_{12}\Sigma_{22}^{-1/2}\right)\mathbf{d} \leq \\ &\left(\mathbf{c}^T\Sigma_{11}^{-1/2}\Sigma_{12}\Sigma_{22}^{-1/2}\Sigma_{22}^{-1/2}\Sigma_{21}\Sigma_{11}^{-1/2}\mathbf{c}\right)^{1/2}(\mathbf{d}^T\mathbf{d})^{1/2},\end{aligned} \tag{3}$$

we have

$$\rho \leq \frac{\left(\mathbf{c}^T\mathbf{\Sigma}_{11}^{-1/2}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1/2}\mathbf{\Sigma}_{22}^{-1/2}\mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1/2}\mathbf{c}\right)^{1/2}}{(\mathbf{c}^T\mathbf{c})^{1/2}}. \tag{4}$$

By defining

$$\mathbf{\Omega} = \mathbf{\Sigma}_{11}^{-1/2}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1/2}, \tag{5}$$

Eqn. (4) is simplified as

$$\rho \leq \frac{\left(\mathbf{c}^T\mathbf{\Omega}\mathbf{c}\right)^{1/2}}{(\mathbf{c}^T\mathbf{c})^{1/2}}. \tag{6}$$

Here we aim to get the maximal value of ρ, so we can maximize the right hand side w.r.t. $\mathbf{c}$. Thus we can make $\mathbf{c}$ the leading eigenvector of $\mathbf{\Omega}$. When $\mathbf{c}$ is the leading eigenvector with largest eigenvalue λ_1 of $\mathbf{\Omega}$, we have

$$\begin{aligned}\rho &\leq \frac{\left(\mathbf{c}^T\mathbf{\Omega}\mathbf{c}\right)^{1/2}}{(\mathbf{c}^T\mathbf{c})^{1/2}} \\ &= \frac{(\mathbf{c}^T\lambda_1\mathbf{c})^{1/2}}{(\mathbf{c}^T\mathbf{c})^{1/2}} = \sqrt{\lambda_1}.\end{aligned} \tag{7}$$

Thus the maximal value of the correlation ρ, *i.e.* ρ_m, is the square root of λ_1. Hence we have

$$\rho_m = \sqrt{\lambda_1}, \tag{8}$$

Putting these pieces together, we reach

$$\tau(\mathcal{F}_1, \mathcal{F}_2) = \sqrt{\max\{\mathrm{eig}(\mathbf{\Sigma}_{11}^{-1/2}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1/2})\}}. \tag{9}$$

To calculate the value of τ, we need to calculate the maximal eigenvalue of $\mathbf{\Omega}$. We notice that this calculation can be simplified by applying Singular Value Decomposition (SVD), since $\mathbf{\Omega}$ can be decomposed as $\mathbf{\Omega} = \mathbf{\Omega}_h\mathbf{\Omega}_h^T$, where $\mathbf{\Omega}_h = \mathbf{\Sigma}_{11}^{-1/2}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1/2}$. Thus we have

$$\begin{aligned}\tau(\mathcal{F}_1, \mathcal{F}_2) =& \max\left(\mathrm{diag}(\mathbf{S})\right), \\ &\text{for } \mathbf{U}, \mathbf{S}, \mathbf{V}^T = \mathrm{svd}(\mathbf{\Sigma}_{11}^{-1/2}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1/2}).\end{aligned} \tag{10}$$

Then we obtain the task similarity of $\mathcal{T}_1$ and $\mathcal{T}_2$ as

$$\begin{aligned}\tau(\mathcal{T}_1, \mathcal{T}_2) &\triangleq \tau(\mathcal{F}_1, \mathcal{F}_2) \\ &= \max\{\mathrm{singular}(\mathbf{\Sigma}_{11}^{-1/2}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1/2})\},\end{aligned} \tag{11}$$

where $\mathrm{singular}(\cdot)$ denotes the operation of getting the singular values in the SVD operation. We can clearly see that the similarity between $\mathcal{T}_1$ and $\mathcal{T}_2$, $\tau(\mathcal{T}_1, \mathcal{T}_2)$, is reflected by the covariance matrices and cross-covariance matrix of $\mathcal{F}_1$ and $\mathcal{F}_2$. The task similarity has good interpretability. The value of the similarity ranges from 0 to 1, where 0 means two tasks are not related, and 1 means they are highly related. Also we notice the following property:

Proposition 1. *Task similarity defined in Eqn.* (11) *is commutative.*

The proposition above agrees with the common sense that the similarity of two tasks is invariant to their orders. The proposition can be proofed mathematically as follows:

Proof.

With the definition of task similarity, we have

$\tau(\mathcal{T}_1, \mathcal{T}_2) = \max\{\mathrm{diag}(\mathbf{S})\}$, for $\mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}} = \mathbf{\Sigma}_{11}^{-1/2}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1/2}$.

Taking the transpose operation, we have

$\mathbf{V}\mathbf{S}^T\mathbf{U}^T = \mathbf{\Sigma}_{22}^{-1/2}\mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1/2}$.

As a result,

$\tau(\mathcal{T}_2, \mathcal{T}_1) = \max\{\mathrm{diag}(\mathbf{S}^{\mathrm{T}})\} = \max\{\mathrm{diag}(\mathbf{S})\} = \tau(\mathcal{T}_1, \mathcal{T}_2)$.

So we have $\tau(\mathcal{T}_1, \mathcal{T}_2) = \tau(\mathcal{T}_2, \mathcal{T}_1)$.

□

We also note the metric in Eqn. (11) has no restriction on the dimensions of $\mathcal{F}_1$ and $\mathcal{F}_2$, thus it is able to measure the similarity of the tasks in both homogeneous and heterogeneous multi-task learning.

In the formulations above, the features vectors $\mathcal{F}_1$ and $\mathcal{F}_2$ are output features and the corresponding similarity metric is the proposed STR. We can make two simple extensions to the formulation as follows. Firstly, when we use the ground truth labels as the feature vectors, the corresponding similarity metric is the ground truth task similarity for $\mathcal{T}_1$ and $\mathcal{T}_2$:

$$\tau^{\mathrm{gt}}(\mathcal{T}_1, \mathcal{T}_2) \triangleq \tau(\mathcal{Y}_1, \mathcal{Y}_2). \tag{12}$$

This ground truth task similarity metric is model-independent, and it serves as an additional supervision signal in our proposed TRN. Secondly, when we use hidden features as the feature vectors, the corresponding similarity metric reflects how the hidden features are related.

3.2. Task Relation Layer and Task Relation Net

The definition of STR is based on the covariance matrices, which can be estimated from all the samples in an off-line fashion. However, the off-line estimation makes STR infeasible when training a deep network, as the features are dynamic during training. In this subsection, we introduce TRL, which estimates the STR from the dynamic features in a deep neural network. Formally, the TRL takes as inputs two random vectors $\mathcal{F}_1$ and $\mathcal{F}_2$ in a mini-batch, estimates the covariance matrices, and outputs the similarity metric between them according to Eqn. (10).

Then we introduce how to build a TRN with TRLs as follows. We use TRLs to enforce constraints, *i.e.* regularizations, on the output features and the hidden features as illustrated in Fig. 1. The TRN adopts a commonly used multi-task learning architecture, where there is a trunk network to learn the shared representations, and two task-specific subnets performing two tasks. We apply TRL between the corresponding layers in the task-specific subnets. Specifically, we can place a TRL between the output layers in the task-specific subnets to form an output TRL. Taking as input the features from output layers, the output TRL generates STR. TRL can also be placed between the labels of the two tasks to form a label TRL, and taking as input the labels, the label TRL outputs the ground truth task similarity. We take this ground truth task similarity to regularize the layer outputs between two tasks. We can also place a TRL between the hidden layers to form a hidden TRL to estimate the similarity between hidden features. TRN uses additional losses to encourage the learned features to have the same similarity compared with the ground truth task similarity.

To train a TRN with TRLs, we define loss terms of task relations in addition to the original loss terms as

$$\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2 + \sum_l \lambda_l \mathcal{L}_{ts}^{(l)}, \tag{13}$$

where $\mathcal{L}_1$ and $\mathcal{L}_2$ are the losses for the first and the sec-

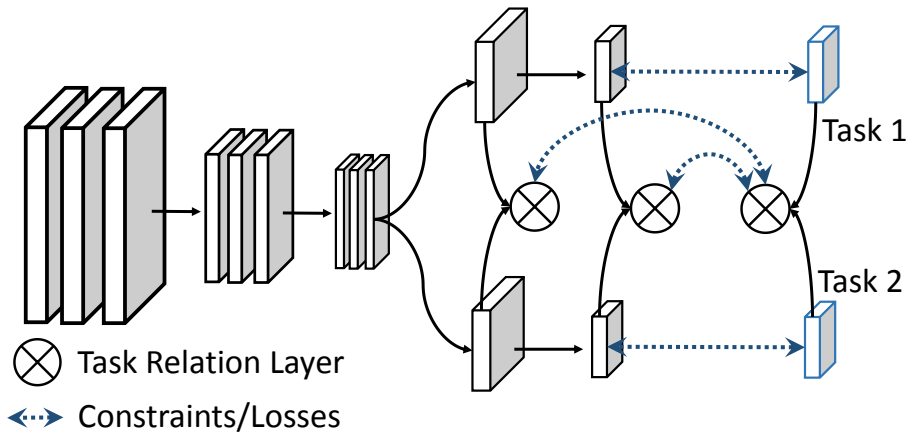

Figure 1. Structure of the proposed TRN containing three TRLs. The three TRLs, from right to left, are label TRL, output TRL and hidden TRL, respectively.

ond task, respectively, in Fig. 1. The term $\mathcal{L}_{ts}^{(l)}$ is the task similarity loss for the l-th layer, with its corresponding loss weight denoted by λ_l. Here l can be the output layer, or the hidden layer. The term $\mathcal{L}_{ts}$ is

$$\mathcal{L}_{ts}^{(l)} = \|\tau(\mathcal{F}_1^{(l)}, \mathcal{F}_2^{(l)}) - \tau(\mathcal{Y}_1, \mathcal{Y}_2)\|, \tag{14}$$

where $\mathcal{F}_1^{(l)}, \mathcal{F}_2^{(l)}$ are the activations of the l-th layer for the first and second task, respectively, and $\tau(\mathcal{F}_1^{(l)}, \mathcal{F}_2^{(l)})$ is the output of the TRL for the l-th layer. Similarly $\tau(\mathcal{Y}_1, \mathcal{Y}_2)$ is the output of the label TRL. With the loss function in Eqn. (13) and an online learning algorithm detailed in the next subsection, all the parameters in a TRN can be learned with standard mini-batch gradient descent.

3.3. Online Learning of TRL

In this subsection, we introduce an online learning algorithm for TRL to capture the similarity of dynamic features when training deep neural networks, as listed in Alg. 1.

The algorithm maintains the global covariance matrices as its parameter. In training it accepts one mini-batch of feature vectors $\{\tilde{\mathbf{F}}_1, \tilde{\mathbf{F}}_2\}$ from one mini-batch of input data during each training step, calculates the covariance matrices for the current mini-batch, and add them to the respective global matrices with a momentum m. Finally the estimated similarity between $\mathcal{F}_1$ and $\mathcal{F}_2$ over the whole dataset is calculated with the updated global covariance matrices. Note that in Alg. 1, the calculation of matrix inverse square root is also realized by SVD as

$$\begin{aligned} &\mathbf{U}, \mathbf{S}, \mathbf{U}^T = \text{svd}(\mathbf{\Sigma}_{11}), \\ &\mathbf{\Sigma}_{11}^{-1/2} = \mathbf{U}\mathbf{S}^{-1/2}\mathbf{U}^T. \end{aligned} \tag{15}$$

Algorithm 1 Online learning algorithm for TRLs

$\mathbf{\Sigma}_{11} \leftarrow \mathbf{I}$
$\mathbf{\Sigma}_{22} \leftarrow \mathbf{I}$
$\mathbf{\Sigma}_{12} \leftarrow \mathbf{0}$ ▷ Initial values of the global covariance matrices
Required: momentum m, batch size
procedure TRL ONLINE($\tilde{\mathbf{F}}_1, \tilde{\mathbf{F}}_2$) ▷ The input features of a mini-batch
  $\tilde{\mathbf{f}}_{1m} \leftarrow \tilde{\mathbf{F}}_1 - \text{mean}(\tilde{\mathbf{F}}_1)$
  $\tilde{\mathbf{f}}_{2m} \leftarrow \tilde{\mathbf{F}}_2 - \text{mean}(\tilde{\mathbf{F}}_2)$ ▷ The demeaned input features of a mini-batch along each feature dimension
  $\tilde{\mathbf{\Sigma}}_{11} \leftarrow \tilde{\mathbf{f}}_{1m} * \tilde{\mathbf{f}}_{1m}^T$
  $\tilde{\mathbf{\Sigma}}_{22} \leftarrow \tilde{\mathbf{f}}_{2m} * \tilde{\mathbf{f}}_{2m}^T$
  $\tilde{\mathbf{\Sigma}}_{12} \leftarrow \tilde{\mathbf{f}}_{1m} * \tilde{\mathbf{f}}_{2m}^T$ ▷ Estimated covariance matrices from one mini-batch of feature vectors
  $\mathbf{\Sigma}_{11} \leftarrow m * \mathbf{\Sigma}_{11} + (1-m) * \tilde{\mathbf{\Sigma}}_{11}$
  $\mathbf{\Sigma}_{22} \leftarrow m * \mathbf{\Sigma}_{22} + (1-m) * \tilde{\mathbf{\Sigma}}_{22}$
  $\mathbf{\Sigma}_{12} \leftarrow m * \mathbf{\Sigma}_{12} + (1-m) * \tilde{\mathbf{\Sigma}}_{12}$ ▷ Update the global covariance matrices with a moment m
  $\tau \leftarrow \max\{\text{singular}(\mathbf{\Sigma}_{11}^{-1/2}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1/2})\}$ ▷ obtain τ with the updated global matrices
  return τ

4. Experiments

We present experiments on two sets of multi-task learning problems, including one homogeneous and one heterogeneous multi-task learning, to demonstrate the effectiveness of the proposed task relation nets for face landmark localization and face attribute classification.

4.1. Experimental Settings

The face landmark localization task aims to predict locations, *i.e.* coordinates, of the key points on facial images,

Figure 2. Face landmark localization and face pose estimation. White dots on faces represent landmark locations, and different columns show faces of different out-of-plane rotation angles. Face landmark localization aims to predict the coordinates of the landmarks, and face pose estimation aims to classify pose angles into discretized bins. We can see that these two tasks are closely related to each other.

such as corners of mouth and center of eyes. The landmark localization task is related to face pose estimation task, as illustrated in Fig. 2. Hence in this experiment, we explore how the prediction of face pose and landmark locations can help boost the performance of each other with the proposed TRN. Note that this is a case of heterogeneous multi-task learning, as the output spaces are of different dimensions.

For the task of face attribute classification, we investigate whether the proposed TRN can facilitate predicting certain face attributes with the help of other attributes. It is a homogeneous multi-task learning, as all the tasks involved are binary classification.

4.1.1 Datasets

The MTFL dataset [25] provides annotations of facial landmark locations for 5 landmarks (10 coordinates), and annotation for face pose angles (discretized into 5 bins). We conduct experiments on multi-task learning of facial landmark localization (a regression task) and face pose classification (a 5-way classification task). The CelebA dataset [14] contains about 200k face images with annotations for 40 attributes, and classifying each attribute corresponds to a binary classification task. There are two versions for the dataset, *i.e.* a pre-cropped version and an original image version, and we use the pre-cropped version in our experiments.

4.1.2 Evaluation Metrics

The evaluation metric for face landmark localization is the widely used normalized mean error [25]. It is defined as the mean discrepancies between the estimated landmark locations and the ground truth, normalized by the inter-ocular distance. The metric for face pose estimation is accuracy. For the CelebA dataset, the evaluation metrics are error rate and average precision.

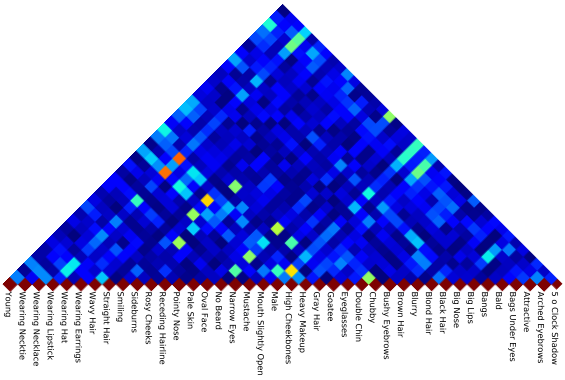

Figure 3. Similarity measurement on the ground truth labels for all pairs of face attributes in the CelebA dataset. Warmer colors (like red, orange) indicate higher similarity values and cooler colors (*e.g* green, blue) indicate lower similarity values.

4.1.3 Baselines

In the experiments, we consider the following baseline methods. The first baseline is the Single Task Learning (STL), where each task is performed by a separate model without weight sharing. The second is Multi-Task Learning (MTL), which is essentially the proposed TRN without TRL. Other baselines are the state-of-the-art methods in the literature, *i.e.* [25, 4] for the MTFL dataset and [14, 19, 11, 3] for CelebA.

4.2. Experiments and Analysis

4.2.1 Validation of the Similarity Metric

To validate the correctness of the similarity measurement, we perform experiments on the CelebA dataset. We compute the ground truth task similarity $\tau(\mathcal{Y}_1, \mathcal{Y}_2)$ for all pairs of 40 face attributes, and the results are given in Fig. 3. In this figure, the pairs of face attributes with high values of τ^{gt} are *Heavy Makeup* and *Wearing Lipstick*, *Wearing Lipstick* and *Male*, *Sideburns* and *No Beard*, *Smiling* and *Mouth Slightly Open*, while those with low values of τ^{gt} are *Receding Hairline* and *Black Hair*, *Rosy Cheeks* and *Narrow Eyes*, *Bangs Under Eyes* and *Wearing Hat*. We can see that the measurement of task similarity agrees well with common sense on whether these two tasks are similar or not.

4.2.2 Convergence of the Online Learning Algorithm

In this subsection, we investigate whether the online learning algorithm (Alg. 1) is capable of converging to the offline calculated ground truth similarity values. We conduct an experiment on two classification tasks in the CelebA dataset. We train two separate models using residual networks [8] with 18 layers (ResNet-18) to perform the two classification

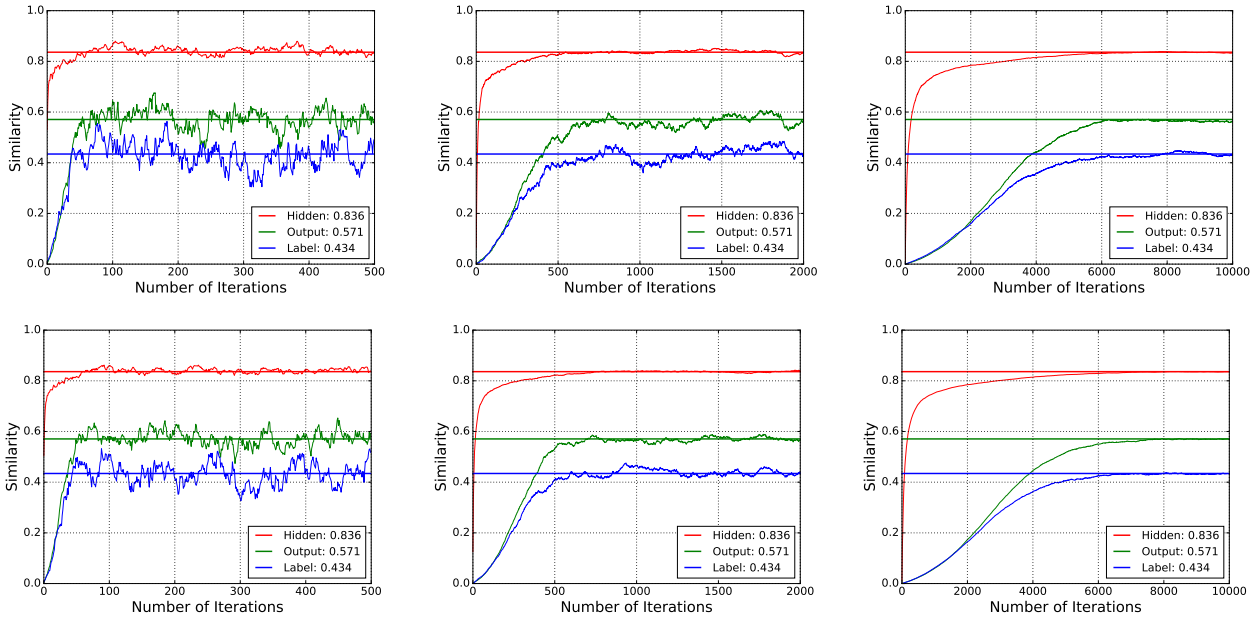

Figure 4. Learning curve of Alg. 1 with different momentum m and batch size. In all figures, the x-axis is the number of training iteration, and the y-axis is the similarity value. The horizontal lines denote the offline calculated similarity values, and the curves show the online learned ones. Row 1: batch size $= 32$, Row, 2: batch size $= 128$. Col. 1-3, $m = 0.9$, 0.99 and 0.999.

tasks. After training, we extract the output features from the output layers and the hidden features from the penultimate layers from the two models. Then we calculate the similarity of the output features, hidden features and labels (given in CelebA) between the two tasks offline using Eqn. (12). Then we use Alg. 1 to online learn the three similarity values with various batch sizes and momentum settings. The results of the online learned similarity values are shown in Fig. 4. We can see that the online learned similarities converge to the offline calculated values for different batch sizes and momentum settings. The convergence behaviors for large batch size (128) and small batch size (32) are similar, but using large batch size is more stable. For momentum, the curve is smoother with high values than low ones, with the cost of slower convergence speed (the scale of x-axis is different in different columns). In the following experiments, we use a batch size of 128 and momentum of 0.99.

We find that the online learning algorithm is able to converge even if the number of channels of feature vectors (512 for hidden feature vectors) is larger than the batch size (128 or 32). We also observe that there is a gap between the similarity of output features and that of labels.

4.2.3 Components Studies in TRN

In this experiment, we demonstrate that the regularization $\mathcal{L}_{\text{ts}}$ in Eqn. (13) contributes to performance improvement on the MTFL dataset. We compare the performance of STL, MTL, TRN (with different settings), and other state-of-the-art methods in facial landmark localization and face pose classification. For STL, the network architecture is ResNet-18 with 5 stages. For MTL, we also adopt ResNet-18, where the trunk contains the first 4 stages, and each task-specific subnet contains the last stage and the final output layer. For TRN, we apply regularizations at different layers, including the output layer and the hidden layer, *i.e.* the hidden features of the last stage after global average pooling. The loss weight λ for the output layer and the hidden layer is 0.01 and 0.001, respectively.

The results of different models are shown in Tab. 1. For STL, different tasks use separate models, and they can explore the feature space independently. Thus the output feature similarity is not far from the ground truth task similarity. When we change the structure of the model from STL to MTL by sharing quite a number of parameters in the trunk network, the similarities of both the hidden features and output features in task-specific layers of subnets have increased, although the MTL still benefits both tasks through multi-task learning. In TRN where explicit modeling of the similarity is used, the output similarity is closer to the ground truth, and the performance is also improved by such modeling. TRN further benefits the tasks when taking the hidden layer into the modeling process.

Model	τ^{hidden}	τ^{output}	Err. (%)	Acc. (%)
[25]	-	-	8.0	-
[4]	-	-	-	75.7
STL	0.84	0.67	7.87	78.70
MTL	0.90	0.69	6.36	79.33
TRN (l=output)	0.88	0.68	6.12	79.60
TRN (l=hidden+output)	0.81	0.66	6.03	80.37

Table 1. Performance of different models on the test set of MTFL. τ^{output} refers to the output task similarity, and τ^{hidden} denotes the similarity of the hidden features. Err. is the normalized mean error of facial landmark localization and Acc. is the accuracy of face pose classification. The ground truth task similarity τ^{gt} is 0.66.

4.2.4 Performance on the CelebA Dataset

We then proceed to evaluate the proposed TRN on CelebA to perform classification of all the 40 attributes. We use the first 5 stages in ResNet-18 as the trunk network with 40 classifier layers on top of the trunk for the MTL baseline. To model the relations of the 40 tasks in TRN, we group the 40 attributes into two groups, and model the relations between the two groups. The first 20 attributes are in the first group and the remaining 20 attributes are in the second group. The loss weight for the task similarity loss is 0.01. We compare the performance of MTL, TRN and other state-of-the-art methods on CelebA dataset, shown in Tab. 2. We can see that our proposed TRN outperforms MTL, and also other state-of-the-art methods in terms of mean Average Precision. TRN also reduces the mean error rate compared to MTL and achieves state-of-the-art performance.

Model	Mean Err. (%)	Mean AP(%)
[19]	9.06	-
[3]	8.67	-
[11]	9.49	77.69
[11]+ Seg	8.20	81.45
MTL	8.42	80.72
TRN	8.21	81.62

Table 2. Results of mean error and mean Average Precision (AP) over all 40 attributes on the test set of CelebA. Here [Kalayeh *et al.*, 2017] is their baseline method and [Kalayeh *et al.*, 2017]+Seg is their final model, which uses additional segmentation information.

4.2.5 TRN for Tasks of Various Similarities

In this experiment, we investigate how TRN behaves for tasks with different task similarities. We choose different pairs of attributes that have varying levels of ground truth task similarities in CelebA, and compare the performance of STL, MTL, and TRN for each pair. Here STL adopts ResNet-18, MTL uses ResNet-18 with two classifier layers, and TRN adds regularization to the output layers of MTL. The chosen pairs and the performance are in Tab. 3. We can observe that the proposed TRL can model the mutual learning of tasks with different ground truth task similarities. For some pairs, where MTL outperforms STL, our proposed TRN can further improve the performance. See Rows 1, 2 and 3 of Tab. 3. For some pairs, where MTL underperforms STL (so-called negative transfer), TRN can improve the performance of MTL to mitigate negative transfer, like in Rows 4 and 5.

Att_1	Att_2	τ^{gt}	AP of Att_1(%)			AP of Att_2 (%)		
			STL	MTL	TRN	STL	MTL	TRN
W.H.	B.	0.031	91.66	92.24	**93.04**	75.32	75.82	**76.98**
Mt.	G.	0.451	56.61	58.54	**59.13**	72.35	72.65	**72.76**
S.B.	N.B.	0.543	74.18	76.21	**76.82**	99.60	99.61	**99.62**
S.B.	A.E.	0.116	**75.85**	74.43	75.82	75.64	75.30	**75.74**
M.	B.E.	0.301	99.54	99.54	**99.55**	59.97	59.44	**60.14**

Table 3. Performance for different pairs of attributes with various τ^{gt}. The short-hand attribute names are: W.H.: *Wearing Hat*, B.:*Bald*, S.B.:*Sideburns*, A.E.:*Arched Eyebrows*, M.:*Male*, B.E.:*Bags Under Eyes*, Mt.:*Mustache*, G.:*Goatee*, N.B.:*No Beard*.

5. Conclusion

In this paper, we propose a quantitative metric named Statistical Task Relation (STR) to measure the relations between tasks in multi-task learning. We develop a Task Relation Layer (TRL) and an online learning algorithm to calculate the STR from dynamic features in a deep neural network. The resultant Task Relation Net (TRN) is able to perform explicit analysis and reasoning with task relation, and better leverage the mutual information of different computer vision tasks. Experiments show that the proposed TRNs are effective for improving the performance for both homogeneous and heterogeneous multi-task learning for vision tasks.

Acknowledgement

The work was supported by the National Research Foundation, Prime Minister's Office, Singapore under its Strategic Capability Research Centres Funding Initiative. The work of Jiashi Feng was partially supported by NUS IDS R-263-000-C67-646, ECRA R-263-000-C87-133, and MOE Tier-II R-263-000-D17-112.

References

[1] R. Caruana. Multitask learning. *Mach. Learn.*, 28(1):41–75, July 1997.

[2] C. Ciliberto, Y. Mroueh, T. Poggio, and L. Rosasco. Convex learning of multiple tasks and their structure. In *International Conference on Machine Learning*, pages 1548–1557, 2015.

[3] H. Ding, H. Zhou, S. K. Zhou, and R. Chellappa. A deep cascade network for unaligned face attribute classification. *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[4] M. Ehrlich, T. J. Shields, T. Almaev, and M. R. Amer. Facial attributes classification using multi-task representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 47–55, 2016.

[5] Y. Fang, Z. Ma, Z. Zhang, X.-Y. Zhang, and X. Bai. Dynamic multi-task learning with convolutional neural network. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 1668–1674, 2017.

[6] H. Han, A. K. Jain, S. Shan, and X. Chen. Heterogeneous face attribute estimation: A deep multi-task learning approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99), 2017.

[7] K. Hashimoto, C. Xiong, Y. Tsuruoka, and R. Socher. A joint many-task model: Growing a neural network for multiple nlp tasks. *arXiv preprint arXiv:1611.01587*, 2016.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[9] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

[10] J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong. Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 7304–7308. IEEE, 2013.

[11] M. M. Kalayeh, B. Gong, and M. Shah. Improving facial attribute prediction using semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[12] Z. Kang, K. Grauman, and F. Sha. Learning with whom to share in multi-task feature learning. In *International Conference on Machine Learning*, pages 521–528, 2011.

[13] J. Li, S. Xiao, F. Zhao, J. Zhao, J. Li, J. Feng, S. Yan, and T. Sim. Integrated face analytics networks through cross-dataset hybrid training. In *Proceedings of the 25th ACM international conference on Multimedia*. ACM, 2017.

[14] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738, 2015.

[15] M. Long, Z. Cao, J. Wang, and S. Y. Philip. Learning multiple tasks with multilinear relationship networks. In *Advances in Neural Information Processing Systems*, pages 1593–1602, 2017.

[16] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3994–4003, 2016.

[17] R. Ranjan, V. M. Patel, and R. Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[18] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[19] E. M. Rudd, M. Günther, and T. E. Boult. Moon: A mixed objective optimization network for the recognition of facial attributes. In *European Conference on Computer Vision*, pages 19–35. Springer, 2016.

[20] S. Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[21] W. Wang, S. J. Pan, and D. Dahlmeier. Multi-task memory networks for category-specific aspect and opinion terms co-extraction. *arXiv preprint arXiv:1702.01776*, 2017.

[22] Y. Yang and T. Hospedales. Deep multi-task representation learning: A tensor factorisation approach. *arXiv preprint arXiv:1605.06391*, 2016.

[23] J. Yu and J. Jiang. Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 236–246, 2016.

[24] Y. Zhang and D.-Y. Yeung. A convex formulation for learning task relationships in multi-task learning. *arXiv preprint arXiv:1203.3536*, 2012.

[25] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Facial landmark detection by deep multi-task learning. In *European Conference on Computer Vision*, pages 94–108. Springer, 2014.