Learning with Whom to Share in Multi-task Feature Learning

Zhuoliang Kang ZKANG@USC.EDU
Department of Computer Science, U. of Southern California, Los Angeles, CA 90089

Kristen Grauman GRAUMAN@CS.UTEXAS.EDU
Department of Computer Science, U. of Texas, Austin, TX 78701

Fei Sha FEISHA@USC.EDU
Department of Computer Science, U. of Southern California, Los Angeles, CA 90089

Abstract

In multi-task learning (MTL), multiple tasks are learnt jointly. A major assumption for this paradigm is that all those tasks are indeed related so that the joint training is appropriate and beneficial. In this paper, we study the problem of multi-task learning of shared feature representations among tasks, while simultaneously determining "with whom" each task should share. We formulate the problem as a mixed integer programming and provide an alternating minimization technique to solve the optimization problem of jointly identifying grouping structures and parameters. The algorithm monotonically decreases the objective function and converges to a local optimum. Compared to the standard MTL paradigm where all tasks are in a single group, our algorithm improves its performance with statistical significance for three out of the four datasets we have studied. We also demonstrate its advantage over other task grouping techniques investigated in literature.

1. Introduction

Multi-task learning (MTL) is a learning paradigm where multiple tasks are jointly learnt (Caruana, 1997; Thrun & Pratt, 1998). The basic notion is that learning one task would benefit from learning other tasks, if they are related. Multi-task learning has been applied to many problems, including those in computer vision (Torralba et al., 2007; Loeff & Farhadi, 2008; Quattoni et al., 2008), natural language processing (Ando & Zhang, 2005), and genomics (Obozinski et al., 2009).

There have been two main ways to define task relatedness. The first one is to assume that the parameters used by all tasks are close to each other, either measured in the Frobenius norms of their differences (Evgeniou & Pontil, 2004; Liu et al., 2009; Zhang & Yeung, 2010; Parameswaran & Weinberger, 2010), or sharing a common prior (Yu et al., 2005; Lee et al., 2007; Daumé, 2009).

The other way to model task relatedness is to assume that all tasks share a common yet latent feature representation (Caruana, 1997; Ando & Zhang, 2005). Argyriou et al. proposed a framework to learn shared features with convex optimization (Argyriou et al., 2008a). Concretely, by forming a parameter matrix with all the parameters of the tasks, their formulation minimizes empirical risk of all tasks, but balanced with a trace-norm based regularizer on the parameter matrix. The trace norm formulation is closely connected to group LASSO (Meier et al., 2008), where the goal is to discover groups of variables that are relevant to prediction tasks only when they are used jointly. In fact, It has been shown that joint covariate selection and subspace selection, a special case of group LASSO, converges to the trace-norm regularization even though the former uses $l_{1,2}$ norm on the parameter matrix (Obozinski et al., 2009).

Being orthogonal to how to define relatedness, another important assumption made by most MTL techniques is that all the tasks are indeed related and appropriate for joint training (or, at least, that an expert using the method can determine those that are). When this assumption does not hold, negative transfer occurs,

Appearing in *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

where learning jointly with dissimilar or outlier tasks results in worsened performance. Though a crucial issue in applying MTL, very few has considered how to address it. Jacob et al. consider the problem of automatically clustering tasks, where tasks in the same cluster are more related to each other than to tasks in a different cluster. In their work, the task relatedness is modeled as similarity in parameters. Dissimilar tasks are then placed in different clusters and learnt independently from each other. Similar ideas were also explored in an earlier work (Thrun & O'Sullivan, 1998), where the similarity between two tasks is quantified by "fitness" – how well one task's parameters perform in the other task.

In this paper, we study the problem of clustering task in the distinct setting where relatedness is modeled as learning shared features among the tasks. The goal is for the algorithm to simultaneously determine "with whom" each task should share features, while also optimizing the model parameters for all tasks per group. Central to our strategy is the observation that *similarity* among classes does not imply that successful sharing can occur between them; rather, the grouping we seek should directly account for the *discriminative task* of interest.

We formulates the problem as a mixed integer programming problem where binary indicator variables are used to assign tasks to groups. Within each group, the tasks share a common feature representation and the parameters are jointly learnt and regularized with trace norms. We provide an alternating minimization technique to solve the optimization problem of jointly identifying grouping structures and parameters. The algorithm monotonically decreases the objective function and converges to a local optimum.

We validate our approach empirically with one synthetic and three realistic datasets. Though it is not guaranteed that learning task grouping structures will always improve the baseline approach (i.e., where all tasks are in a single group), our algorithm improves the performance with statistical significance for three out of the four datasets. In addition, we demonstrate its advantage over several existing task clustering techniques. In particular, we contrast our work to a recent technique proposed in (Argyriou et al., 2008b), where the problem of inferring task clusters is cast as learning a set of kernel functions, one for each group.

We also show how our method can incorporate grouping structures from previously learned tasks into new tasks with potentially different grouping structures, and demonstrate empirically that exploiting "old structures" can help accuracy on new tasks.

The paper is organized as follows. In section 2, we describe the standard framework of multi-task feature learning. In section 3, we describe our formulation and optimization techniques for jointly inferring optimal task grouping structures and parameters of all the tasks. We present results from the empirical studies in section 4. We conclude and discuss future research directions in section 5.

2. Multi-task Feature Learning

In what follows, we describe briefly the framework of multi-task feature learning (MTFL) (Argyriou et al., 2008a). Our work builds directly on their work and will be described in section 3.

We assume that there are T supervised learning tasks. Let $\ell(\mathcal{D}_t; \boldsymbol{w}_t)$ stand for the loss function of the t-th task on its training data $\mathcal{D}_t$, and $\boldsymbol{w}_t$ be the corresponding model parameter vector. We assume that all tasks use the same feature space, with the feature vector denoted by $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^{\mathsf{D}}$. In the standard learning paradigm, the tasks are learned independently, sharing no information among them,

$$\boldsymbol{W}^* = \arg\min \sum_t \ell(\mathcal{D}_t; \boldsymbol{w}_t) + \gamma \|\boldsymbol{W}\|_F^2 \qquad (1)$$

where the parameter matrix $\boldsymbol{W}$ is composed of $\{\boldsymbol{w}_t\}$ as column vectors. $\|\boldsymbol{W}\|_F^2$ denotes the squared Frobenius form and decomposes into the familiar ℓ_2 regularization on each $\boldsymbol{w}_t$: $\|\boldsymbol{W}\|_F^2 = \sum_t \|\boldsymbol{w}_t\|_2^2$.

In MTFL, we assume that there is a shared feature subspace $\mathcal{U}$ on which all tasks perform well. One goal of MTFL is to identify this subspace. In the simplest setting, the feature subspace is a linear transformation of the original "raw" feature vector: $\boldsymbol{u}_n = \boldsymbol{U}^{\mathrm{T}} \boldsymbol{x}_n$, where $\boldsymbol{U} \in \mathbb{R}^{\mathsf{D} \times \mathsf{D}}$ is an orthogonal matrix. Moreover, we consider linear classifiers in the space of $\mathcal{U}$, where the discriminant function is given by

$$\boldsymbol{\theta}_t^{\mathrm{T}} \boldsymbol{u}_n = \boldsymbol{\theta}_t^{\mathrm{T}} \boldsymbol{U}^{\mathrm{T}} \boldsymbol{x}_n = \boldsymbol{w}_t^{\mathrm{T}} \boldsymbol{x}_n. \qquad (2)$$

In other words, we aim to search for parameters $\boldsymbol{w}_t$ such that they are linear combinations of a common set of basis $\boldsymbol{w}_t = \boldsymbol{U}\boldsymbol{\theta}_t$. Such representation is certainly not unique without further constraints. To this end, MTFL enforces the low-rank constraint, that is, there are only a few basis in $\boldsymbol{U}$ that are necessary in order to compose all $\boldsymbol{w}_t$. This is achieved by regularizing $\boldsymbol{\Theta}$, whose columns are $\boldsymbol{\theta}_t$, with $(2,1)$-norm,

$$\boldsymbol{\Theta}^*,\ \boldsymbol{U}^* = \arg\min \sum_t \ell(\mathcal{D}_t; \boldsymbol{\theta}_t^{\mathrm{T}}\, \boldsymbol{U}^{\mathrm{T}}) + \gamma \|\boldsymbol{\Theta}\|_{2,1}^2 \qquad (3)$$

The norm is given by $\|\boldsymbol{\Theta}\|_{2,1} = \sum_{d=1}^{D} \sqrt{\sum_t \theta_{dt}^2}$. An

important property of this norm is that it computes the 2-norm of parameter values in each dimension *across* tasks. Consequently, for any dimension d, the regularization attains the minimum if and only if the corresponding parameters are all zero: $\theta_{dt} = 0$ for all t. Therefore, the regularization would choose the $\mathbf{\Theta}$ with the *smallest* number of *non-zero rows*. This is equivalent to finding a subset of feature dimensions in $\mathcal{U}$ that are useful for all tasks, namely, whose corresponding parameters are nonzero.

The optimization algorithm for eq. (3) starts by identifying it with the following equivalent form

$$\begin{aligned} \boldsymbol{W}^*,\ \mathbf{\Omega}^* = \arg\min \sum_t \ell(\mathcal{D}_t; \boldsymbol{w}_t) \\ + \gamma \sum_t \boldsymbol{w}_t^{\mathrm{T}} \mathbf{\Omega}^{-1} \boldsymbol{w}_t + \gamma\epsilon\, \mathsf{Trace}(\mathbf{\Omega}^{-1}), \end{aligned} \tag{4}$$

where $\mathbf{\Omega} \in \mathbb{R}^{\mathsf{D}\times\mathsf{D}}$ is constrained to be a positive definite matrix with bounded trace $\mathsf{Trace}\,(\mathbf{\Omega}) \leq 1$. $\epsilon \ll 1$ is a smoothing parameter for numerical stability and benign convergence properties (cf. Theorem 3 in (Argyriou et al., 2008a)).

When ϵ is zero, minimizing $\mathbf{\Omega}$ first leads to a closed form solution in $\boldsymbol{W}$ (Argyriou et al., 2008a). The optimization problem is then reformulated as

$$\boldsymbol{W}^* = \arg\min \sum_t \ell(\mathcal{D}_t; \boldsymbol{w}_t) + \gamma \|\boldsymbol{W}\|_*^2, \tag{5}$$

where $\|\boldsymbol{W}\|_*$ is the trace norm of the parameter matrix. Since trace norm is the convex envelope of matrix rank, minimizing the trace-norm regularizer has the effect of preferring a low-rank solution of $\boldsymbol{W}$.

The above framework of MTFL can also be extended to nonlinear classifiers where the input feature $\boldsymbol{x}$ is mapped to a kernel feature space $\boldsymbol{\phi}(\boldsymbol{x})$. Details can be found in (Argyriou et al., 2008a).

3. Learning to Group Tasks

The multi-task learning framework described in the previous section assumes that all tasks are related. This leads to a single-term regularization on all the parameters. In practice, this assumption could be too restrictive, as illustrated by the following example.

Suppose the feature space is decomposed as the direct sum of two orthogonal-complement subspaces $\mathcal{U} = \mathcal{U}_1 \oplus \mathcal{U}_2$. Associated with each subspace, there are a group of tasks whose parameters depend only on the basis of that subspace. In the previous notation of ours (cf. eq. (2-3)), the parameter matrix $\mathbf{\Theta}$ in the feature space is thus block-diagonal. Thus, regularizing the matrix to have zero rows as much as possible would not be suitable. Particularly, one possible danger would be forcing truly useful rows to be zero, thus damaging the performance of the affected models.

Thus it is desirable to be able to cluster tasks automatically into separate groups where task relatedness is more prominent within each group. Regularization should occur for tasks in the same group and should not be imposed for tasks across groups. In the following, we show how such inference can be achieved by extending the previous work. We start by formulating the problem as an integer programming and then describe optimization techniques for solving it.

3.1. Integer Programming

Let G denote the number of groups into which we want to cluster our tasks. When G is unknown, as often in practical applications, we use cross-validation to select the optimal one. For the g-th group, let $\boldsymbol{W}_g$ denote the parameter matrix for the tasks in this group. These tasks are to be learnt jointly but independently from tasks in other groups. Given the group assignment of tasks, analogous to eq. (5) where all tasks are in a single group, we have the following for separately grouped tasks,

$$\boldsymbol{W}^* = \arg\min \sum_t \ell(\mathcal{D}_t; \boldsymbol{w}_t) + \gamma \sum_g \|\boldsymbol{W}_g\|_*^2. \tag{6}$$

Let $q_{gt} \in \{0, 1\}$ be a binary variable indicating whether the t-th task is assigned to the group g. Let $\boldsymbol{Q}$ be the group assignment matrix composed of q_{gt} as its elements. Let $\boldsymbol{Q}_g \in \mathbb{R}^{\mathsf{T}\times\mathsf{T}}$ be the diagonal matrix whose diagonal elements are q_{gt}. $\|\boldsymbol{W}_g\|_*$ is thus expressed as

$$\|\boldsymbol{W}\boldsymbol{Q}_g\|_* = \mathsf{Trace}\left[\boldsymbol{W}\boldsymbol{Q}_g\,(\boldsymbol{W}\boldsymbol{Q}_g)^{\mathrm{T}}\right]^{1/2} \tag{7}$$

To automatically infer the grouping, we seek both the model parameters $\boldsymbol{W}$ and the group assignment matrices $\boldsymbol{Q}_g$ that minimize

$$\begin{aligned} \min \quad & \sum_t \ell(\mathcal{D}_t; \boldsymbol{w}_t) + \gamma \sum_g \|\boldsymbol{W}\boldsymbol{Q}_g\|_*^2 \\ \text{s.t.} \quad & \sum_g \boldsymbol{Q}_g = \boldsymbol{I} \text{ with } q_{gt} \in \{0, 1\} \end{aligned} \tag{8}$$

where $\boldsymbol{I}$ stands for the identity matrix. The summation constraint ensures that each task is assigned to one and only one group.

The framework of eq. (8) includes two special cases. When $\mathsf{G} = 1$, the framework reduces to the standard multitask feature learning discussed in the section 2 where all tasks belong to a single group. When

$\mathsf{G} = \mathsf{T}$ — the number of tasks, the framework reduces to standard supervised learning where tasks are independently learnt from each other (cf. eq. (1)).

Eq. (8) is a mixed integer programming problem, where finding the global optimum is generally intractable. We show how to solve it in the following.

3.2. Optimization

Our main approach is the technique of alternative minimization for eq. (8). When all $\boldsymbol{Q}_g$ are fixed, eq. (8) can be minimized over $\boldsymbol{W}$ in a similar way as eq. (5). The challenge is to find the optimal $\boldsymbol{Q}_g$ while holding $\boldsymbol{W}$ fixed. In this step, note that only the regularizer $R(\boldsymbol{Q}) = \sum_g \|\boldsymbol{W}\boldsymbol{Q}_g\|_*^2$ is relevant.

There are several strategies we have explored. We have found the following one to be especially effective. Specifically, we use a different regularizer $T(\boldsymbol{Q}) = \sum_g \|\boldsymbol{W}\sqrt{\boldsymbol{Q}_g}\|_*^2$. For binary q_{gt}, the two regularizers are precisely the same. However, with q_{gt} being relaxed to be continuous, the two have different computational properties. Specifically, $T(\boldsymbol{Q})$ is not a convex function in q_{gt}. Nevertheless, the following theorem reveals an interesting fact about $T(\boldsymbol{Q})$.

Theorem 1. *Let $\{\boldsymbol{Q}_g^*\}$ be either the solution or a local optimum to the following optimization,*

$$\begin{aligned} \min \quad & T(\boldsymbol{Q}) = \sum_g \|\boldsymbol{W}\sqrt{\boldsymbol{Q}_g}\|_*^2 \\ \mathsf{s.t} \quad & \sum_g \boldsymbol{Q}_g = \boldsymbol{I} \text{ with } 0 \le q_{gt} \le 1 \end{aligned} \tag{9}$$

then either one of the following is true: i) $\{\boldsymbol{Q}_g^*\}$ is binary; ii) there exists another binary $\{\boldsymbol{Q}'_g\}$ such that $T(\boldsymbol{Q}^*) = T(\boldsymbol{Q}')$.

The proof is presented in the Supplementary Material (Kang et al., 2011). While the theorem does not provide a way to identify the binary $\boldsymbol{Q}'$ from a fractional solution $\boldsymbol{Q}^*$, we encountered very infrequently fractional solutions in our experiments with our nonlinear optimization algorithm to be described in the following.

To handle the constraints in eq. (9) on g_{gt}, we reparameterize them with unconstrained variables α_{gt}

$$q_{gt} = e^{\alpha_{gt}}/Q_0, \quad Q_0 = \sum_g e^{\alpha_{gt}}. \tag{10}$$

We then solve eq. (9) by the method of gradient-descent on α_{gt}. The key step is to compute the gradient of $T(\boldsymbol{Q})$ with respect to $\boldsymbol{Q}_g$; the details are given in the Supplementary Material (Kang et al., 2011).

Since $T(\boldsymbol{Q})$ is not a convex function of α_{gt}, the gradient-descent method could get trapped in local optima. Thus, in practice, we initialize the gradient-descent with 10 sets of random α_{gt} and choose the set that leads to the best (local) optimum. This heuristic works well in our experiments. The algorithm listing in Algorithm 1 illustrates crucial steps of solving eq. (8). The iterative procedure is terminated when the algorithm converges, when there is very little change of either $\boldsymbol{W}$ or $\boldsymbol{Q}$ between two consecutive iterations.

Algorithm 1 Alternative Minimization of Eq. (8)

Input: step size η for gradient-descent
Output: $\boldsymbol{W}^*$, $\boldsymbol{Q}^*$

1: Initialize $\boldsymbol{W}$ and $\boldsymbol{Q}$
2: **while** not converged **do**
3: **for** $g = 1$ to G **do**
4: Solve the following with the Algorithm 1 in (Argyriou et al., 2008a)

$$\min \sum_{t:q_{gt}=1} \ell(\mathcal{D}_t; \boldsymbol{w}_t) + \gamma\|\boldsymbol{W}_g\|_*^2 \tag{11}$$

5: **end for**
6: Fix $\boldsymbol{W}_g$, identify the optimal $\boldsymbol{Q}$

$$\alpha_{gt} \leftarrow \alpha_{gt} - \eta\, \partial T(\boldsymbol{Q})/\partial \alpha_{gt} \tag{12}$$

7: **end while**

3.3. Other Extensions

Nonlinear feature space. The method we have described so far can be easily extended to nonlinear kernel classifiers. The following representer theorem is in parallel to the Theorem 4 in (Argyriou et al., 2008a).

Theorem 2. *Let $\phi(\boldsymbol{x})$ denote the nonlinear feature map by a reproducing kernel. Assume the linear discriminant function $\boldsymbol{w}_t^T\phi(\boldsymbol{x})$ for the t-th task. Then the optimal solution of eq. (8) is given by*

$$\begin{aligned} \boldsymbol{w}_t^* &= \sum_{s=1}^{\mathsf{T}} \sum_{n=1}^{\mathsf{N}} h_{st} c_{sn}^t \phi(\boldsymbol{x}_{sn}) \\ h_{st} &= \sum_{g=1}^{\mathsf{G}} q_{gs}^* q_{gt}^* \end{aligned} \tag{13}$$

where c_{sn}^t is the linear combination coefficients for the t-th task, combining total $\mathsf{N} \times \mathsf{T}$ training data $\boldsymbol{x}_{sn}$.

The proof is straightforward and omitted. Note that the binary variable h_{st} can be interpreted as the element of a kernel matrix, measuring the similarity between the t-th task and the s-th task. Thus, learning the optimal grouping is equivalent to learning a kernel

matrix for characterizing task relatedness, which was explored previously in (Argyriou et al., 2008c).

Transferring to new tasks. Once the parameter matrix $\boldsymbol{W}$ and the optimal groupings $\boldsymbol{Q}_g$ are learnt, they can be used as a "prior" to facilitate the learning process of new tasks. In particular, the new tasks have 3 options: assigned to existing groups of old tasks, clustered into new groups on their own, or a hybrid of both. Our framework in eq. (8) can be readily adapted for this purpose.

Concretely, we increase the number of columns in $\boldsymbol{W}$ to accommodate the new tasks but hold previously learnt parameters (from the "old tasks") fixed. We also increase the dimensions of $\boldsymbol{Q}_g$ to enable creating new groups, but hold previously learnt group assignments for the old tasks fixed. Since parameters of the old tasks are not to be changed, we do not need to include their associated loss functions or their training data in eq. (8). This is often desirable as one might not have access to training data of those tasks, for example, in the scenario of online learning of visual categories. The adapted formulation for incorporating new tasks requires only minor modification to the numerical optimization techniques described in section 3.2. In short, in the Algorithm 1, for parameters and group assignment variables that are fixed due to their association with old tasks, we just set their gradients artificially as zeroes so that they are not updated.

4. Experiments

We present extensive empirical studies to evaluate our approach. For reference, we consider several baselines:

- **Single task:** a baseline single-task learning approach, in which the tasks are learned separately, denoted by $\mathsf{G} = \mathsf{T}$, where T is the number of tasks.
- **No groups-MTL:** the typical MTL approach in which all tasks are learned jointly, indiscriminately putting them into one group, i.e., $\mathsf{G} = 1$.
- **Random groups-MTL:** a naive grouping approach that simply randomly partitions the tasks into groups.
- **Similarity groups-MTL:** a grouping approach that partitions the tasks purely according to their similarity in the original feature space. (We define in more detail below.)
- **Fitness groups-MTL:** a grouping approach that partitions the tasks according to their fitness to each other, by measuring how well one task's parameters can be used *directly* on the other task (Thrun & O'Sullivan, 1998).
- **Kernel groups-MTL:** a grouping approach that assign tasks sequentially and greedily to clusters; within each cluster, the tasks are jointly learnt by using a common kernel function for all tasks' data (Argyriou et al., 2008b). The resulting formulation is similar to eq. (8) except that the trace-norm instead of squared trace-norm is used for regularization. Moreover, online stochastic gradient descent is used to learn the kernel functions and the cluster assignments.

Our goal is to demonstrate that our automatically inferred groups can outperform these methods in many cases. For completeness, we examine several datasets. We start with toy synthetic data, for which task relatedness is well-defined and known *a priori*. Then we present results on two handwritten digit recognition datasets, USPS and MNIST (LeCun et al., 1998; Hull, 1994). Finally, we study the effectiveness of multi-task learning on the Animals with Attributes image dataset (Lampert et al., 2009), testing both our primary automatic grouping algorithm as well as the variant in which we incorporate new tasks.

4.1. Synthetic data

We evaluate on synthetic data as an illustrative sanity check. Our synthetic data consists of 20-dimensional feature vectors, three groups of tasks, and 15 training points per task. Within each group, there are 10 tasks whose parameter vectors are identical to each other up to a scaling. These parameters are used in the model of linear regression to generate training data (inputs and target outputs).

Table 1 displays the root-mean-squared-error (RMSEs) for different numbers of groups, including two of the special baseline cases: $\mathsf{G} = \mathsf{T}$ denotes the case where every single task is learnt separately, and $\mathsf{G} = 1$ denotes the standard MTL approach where all 30 tasks are clustered in a single group. $\mathsf{G} = 1$ clearly outperforms independent learning, confirming the benefit of multi-task learning. However, the best performance is achieved when $\mathsf{G} = 3$, precisely the number of groups that we have used to create the tasks and data.

Fig. 1 displays the matrix $\boldsymbol{Q}$ whose elements are q_{gt}—that is, each entry is an indicator variable for whether the t-th task is assigned to the g-th group. The horizontal axis is t, and the vertical axis is g. The tasks are arranged according to their ground-truth groups (which is information withheld from the learning algorithm). We see that when the number of groups G is

Table 1. RMSE when applying our algorithm on the synthetic dataset, as a function of the number of groups G.

G	T	1	2	3	4
RMSE	0.97	0.48	0.45	**0.42**	0.47

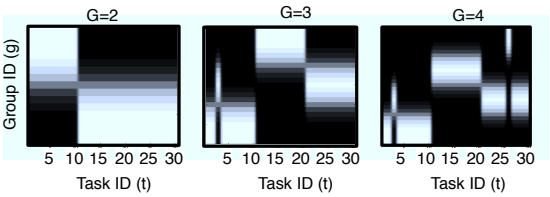

Figure 1. Visualizing how tasks are grouped in the synthetic dataset. Given the correct number of groups $\mathsf{G} = 3$, the correct groupings are discovered for all but one tasks .

set to be 3, our algorithm discovers the true grouping structures of all tasks but one. In contrast, when too few or too many groups are used, the structures either merge or are overly-segmented.

4.2. Handwritten digit recognition

We next study the effectiveness of our approach on handwritten digit recognition using two datasets. Here we treat multi-way classification as a multi-task learning problem, where each task is a classification task of one digit against all the others (Amit et al., 2007). In all experiments in this section, we use logistic regression for the binary classification tasks.

To construct the Similarity groups-MTL baseline for the digits dataset, we explore a couple options. For the USPS data, we project the mean feature vectors of each class into two dimensions, and form groupings by visually examining the proximities of different digit classes on the plane. For the MNIST data, we use the t-SNE algorithm (van der Maaten & Hinton, 2008) to project data into two dimensions, and form groupings again by visually examining clustering and proximities of different classes. This Similarity-groups baseline is intended to serve as a simple but "intelligent" grouping strategy, that is, an educated guess of what classes might share features based on their apparent similarity (as compared to the naive Random grouping baseline). The two options yield similar results on both datasets and we report the best performing ones with the number of groups chosen using the validation data set.

4.2.1. USPS digits dataset

The first digit dataset is the USPS dataset (Hull, 1994). We preprocess the images with PCA, reducing dimensionality to 87 so as to retain $\sim 95\%$ of the total variance. We extract 2000 samples, and use 1000, 500, and 500 samples for training, validation, and test sets, respectively. The test samples are fixed, and we conduct 5 random splits to obtain results across different training and validation samples. We report 10-way misclassification rates of recognizing digits. For all results, we automatically select the number of groups for our approach using the validation set.

Table 2 (middle column) summarizes the results, comparing our method to the baselines defined above. The values represent the mean error rates across 5 splits, and one standard error, defined as the standard deviation of error rates and scaled by $1/\sqrt{5}$.

The results show that conventional MTL—while improving over single-task learning—suffers by requiring all tasks to share features. The Random grouping baseline improves the error slightly, and the Similarity grouping baseline is even a bit better, already showing the influence of splitting tasks that can cause negative transfer. However, the very best results are obtained with our approach (see last row), which determines which tasks should share directly from the data. Interestingly, our method's advantage over the Similarity groups-MTL baseline indicates that simply judging relatedness based on *similarity* is insufficient to identify groups of tasks that enhance *discriminative* power when sharing.

Our approach does not exhibit statistically significant differences from Kernel groups-MTL. We note in passing that this approach requires significant amount of time to learn grouping structures and model parameters. Our experiments (including those on other data sets reported below) indicate it is often one or two-order slower than ours. While heuristic in nature, Fitness groups-MTL is a very strong performer. However, it relies on the assumption that parameters from different tasks are similar in both scales and "semantics" – the tasks are of the same type, i.e, classifying object categories from images of similar data characteristics, etc. Thus, its applicability to learning tasks with different characteristics, for example, classifying object categories versus detecting visual attributes, needs to be studied (Hwang et al., 2011).

4.2.2. MNIST digits dataset

The second digits dataset is the MNIST dataset. We perform preprocessing similarly as for the USPS data, except reducing dimensionality to 64. Table 2 (right column) shows the results. Our method's improvement over standard MTL or the alternate grouping baselines is minor in this case. This could be explained

Table 2. Error rates (%) on two different handwritten digit recognition datasets. The proposed approach outperforms several natural baselines. See text for details.

	USPS	MNIST
Single task ($\mathsf{G} = \mathsf{T}$)	9.5 ± 0.4	15.9 ± 0.3
No groups-MTL ($\mathsf{G} = 1$)	8.8 ± 0.1	15.6 ± 0.3
Random groups-MTL	8.8 ± 0.1	15.4 ± 0.2
Similarity groups-MTL	8.6 ± 0.2	15.4 ± 0.3
Fitness groups-MTL	8.5 ± 0.1	15.2 ± 0.2
Kernel groups-MTL	8.5 ± 0.4	15.7 ± 0.2
Ours	$\mathbf{8.4} \pm 0.3$	$\mathbf{15.2} \pm 0.3$

Table 3. Recognition accuracy (%) on the 20-class Animals dataset.

Single task ($\mathsf{G} = \mathsf{T}$)	No groups- ($\mathsf{G} = 1$)	Fitness-MTL	Ours
26.1 ± 0.2	27.5 ± 0.3	28.0 ± 0.5	$\mathbf{28.4} \pm 0.4$

by the fact that there is very small improvement of $\mathsf{G} = 1$ over $\mathsf{G} = \mathsf{T}$; that is, it seems that multi-task learning is generally not helpful in this data.

On both digit datasets, we found that the two strategies for creating the Similarity-based groups performed similarly, and was indeed relatively close to random grouping. This result lends further evidence that our strategy to learn groups that enable discriminative feature sharing is effective.

4.3. Animal recognition

We next experiment with an image classification task using the Animals with Attributes dataset (Lampert et al., 2009). We use the SIFT bag of word descriptors kindly provided by the dataset creators, which represent each image with a 2000-dimensional histogram over prototype local feature patches. We reduce the dimensionality to 202 using PCA, to retain 95% of the variance. We choose the first 20 animal classes in the data set: *antelope, grizzly-bear, killer-whale, beaver, Dalmatian, Persian-cat, horse, german- shepherd, blue-whale, Siamese-cat, skunk, ox, tiger, hippopotamus, leopard, moose, spider-monkey, humpback-whale, elephant*, and *gorilla*. We use 50, 20, and 30 examples per class for training, validation, and testing, respectively. The tasks are to recognize the different animal classes.

Table 3 shows the recognition accuracy for different methods. There is a clear and significant improvement of using multi-task learning ($\mathsf{G} = 1$) over independent learning ($\mathsf{G} = \mathsf{T}$). Our method's improvement over the standard multi-task learning approach without grouping is also significant. The difference between the mean accuracies is 0.9, significantly greater than the sum of the two standard errors. We did not fully test other approaches (Random groups-MTL and Similarity groups-MTL) because of ineffectiveness in the previous experiments (cf. Table 2) and the extra computational cost. The method of Kernel groups-MTL has an accuracy of 26.3 ± 0.1, significantly lower than other methods.

Table 4. Recognition accuracy (%) out of 100 test examples as a function of the number of training examples N, learning to transfer to the 10-class Animals dataset from the other 20-class Animals dataset described in the text.

N	Single task	Fitness-MTL	Ours
10	31 ± 1.5	37 ± 1.7	$\mathbf{40.6} \pm 1.2$
30	36.6 ± 1.1	39.2 ± 1.6	$\mathbf{39.4} \pm 0.7$
50	39.2 ± 1.3	$\mathbf{42.4} \pm 1.5$	38.9 ± 0.4

Finally, we investigate the proposed variant of our method for incorporating new tasks into an existing group structure we learned previously. To test whether this can be beneficial, we select another 10 animal classes in the dataset. We then use the algorithm described in Section 3.3 to automatically decide whether these 10 classes should be added to the existing grouping structure, form new groups, or both.

Table 4 displays the accuracy of the 10 added classes: independent learning versus 2 transfer learning methods differentiated by how the new 10 tasks are grouped. The result suggests that particularly when there are few training examples, exploiting existing structures for MTL may significantly improve the accuracy. Furthermore, our approach achieves the most significant gain when the number of training examples for the new task is 10. However, this gain diminishes when the number of training examples is increased. In particular, the approach of Fitness-MTL achieves the highest accuracy when $N = 50$, though the difference from our approach is not significant, considering the relatively large standard errors. (Kernel groups-MTL does not perform as well as the other two methods and are thus not reported here.)

5. Conclusion

In this paper, we study the problem of how to partition multiple tasks into groups where within each group, tasks are related and can jointly learn a shared feature representation. Compared to the dominant paradigm where all tasks are assumed to be related, our approach provides the means to deter negative transfer

where learning unrelated and dissimilar tasks jointly results in worsened performance. Empirical studies validated our approach and showed that it is indeed beneficial to simultaneously identifying task grouping structures and task parameters. For future work, we plan to study richer models that can organize tasks in more complicated structures (such as hierarchical trees) than clusters.

References

Amit, Yonatan, Fink, Michael, Srebro, Nathan, and Ullman, Shimon. Uncovering shared structures in multiclass classification. In *Proceedings of the 24th International Conference on Machine learning*, pp. 17–24, 2007.

Ando, Rie Kubota and Zhang, Tong. A framework for learning predictive structures from multiple tasks and unlabeled data. *J. Mach. Learn. Res.*, 6:1817–1853, 2005.

Argyriou, Andreas, Evgeniou, Theodoros, and Pontil, Massimiliano. Convex multi-task feature learning. *Machine Learning*, 73:243–272, 2008a.

Argyriou, Andreas, Maurer, Andreas, and Pontil, Massimiliano. An algorithm for transfer learning in a heterogeneous environment. In *Proc. of ECML/PKDD*, pp. 71–85, 2008b.

Argyriou, Andreas, Micchelli, Charles A., Pontil, Massimiliano, and Ying, Yiming. A spectral regularization framework for multi-task structure learning. In Platt, J.C., Koller, D., Singer, Y., and Roweis, S. (eds.), *Advances in Neural Information Processing Systems 20*, pp. 25–32. MIT Press, Cambridge, MA, 2008c.

Caruana, Rich. Multitask learning. *MLJ*, 28:41–75, 1997.

Daumé, III, Hal. Bayesian multitask learning with latent hierarchies. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pp. 135–142, Arlington, Virginia, United States, 2009. AUAI Press.

Evgeniou, Theodoros and Pontil, Massimiliano. Regularized multi–task learning. In *KDD*, pp. 109–117, 2004.

Hull, J.J. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:550–554, 1994.

Hwang, Sung Ju, Sha, Fei, and Grauman, Kristen. Sharing features between objects and their attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, CO., 2011.

Jacob, L., Bach, F., and Vert, J.P. Clustered multi-task learning: A convex formulation. *Advances in Neural Information Processing Systems*, 21:745–752, 2009.

Kang, Zhuoliang, Grauman, Kristen, and Sha, Fei. Learning with whom to share in multi-task feature learning: Supplementary material, 2011. URL http://www-rcf.usc.edu/~feisha/pubs/icml2011.

Lampert, C.H., Nickisch, H., and Harmeling, S. Learning to detect unseen object classes by between-class attribute transfer. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition,*, pp. 951–958, 2009.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognitio. *Proc. of IEEE*, 86:2278–2324, 1998.

Lee, S.I., Chatalbashev, V., Vickrey, D., and Koller, D. Learning a meta-level prior for feature relevance from multiple related tasks. In *Proceedings of the 24th International Conference on Machine learning*, pp. 496. ACM, 2007.

Liu, J., Ji, S., and Ye, J. Multi-task feature learning via efficient l 2, 1-norm minimization. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pp. 339–348. AUAI Press, 2009.

Loeff, N. and Farhadi, A. Scene Discovery by Matrix Factorization. In *European Conf. Comp. Vis.*, 2008.

Meier, Lukas, van de Geer, Sara, and Bühlmann, Peter. The group LASSO for logistic regression. *Journal Of The Royal Statistical Society Series B*, 70(1):53–71, 2008.

Obozinski, G., Taskar, B., and Jordan, M.I. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, pp. 1–22, 2009.

Parameswaran, Shibin and Weinberger, Kilian. Large margin multi-task metric learning. In Lafferty, J., Williams, C. K. I., Shawe-Taylor, J., Zemel, R.S., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems 23*, pp. 1867–1875. MIT Press, 2010.

Quattoni, Ariadna, Collins, Michael, , and Darrell, Trevor. Transfer learning for image classification with sparse prototype representations. In *Proc. of CVPR*, 2008.

Thrun, Sebastian and O'Sullivan, Joseph. *Clustering learning tasks and the selective cross-task transfer of knowledge*, pp. 235–257. Kluwer Academic Publishers, Norwell, MA, USA, 1998.

Thrun, Sebastian and Pratt, Lorien. *Learning to learn*. Kluwer Academic Publishers, Norwell, MA, USA, 1998. ISBN 0-7923-8047-9.

Torralba, Antonio, Murphy, Kevin P., and Freeman, William T. Sharing visual features for multiclass and multiview object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29:854–869, 2007.

van der Maaten, L.J.P. and Hinton, G.E. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research 9*, pp. 2579–2605, 2008.

Yu, Kai, Tresp, Volker, and Schwaighofer, Anton. Learning gaussian processes from multiple tasks. In *Proceedings of the 22nd international conference on Machine learning*, pp. 1012–1019, New York, NY, USA, 2005. ACM.

Zhang, Y. and Yeung, D.Y. A Convex Formulation for Learning Task Relationships in Multi-Task Learning. In *UAI*, 2010.