Asymmetric Multi-task Learning Based on Task Relatedness and Loss

Giwoong Lee SOPP0002@UNIST.AC.KR
School of Electrical and Computer Engineering, UNIST, Ulsan, South Korea

Eunho Yang EUNHOY@CS.KAIST.AC.KR
School of Computing, KAIST, Daejon, South Korea

Sung Ju Hwang SJHWANG@UNIST.AC.KR
School of Electrical and Computer Engineering, UNIST, Ulsan, South Korea

Abstract

We propose a novel multi-task learning method that minimizes the effect of negative transfer by allowing asymmetric transfer between the tasks based on task relatedness as well as the amount of individual task losses, which we refer to as Asymmetric Multi-task Learning (AMTL). To tackle this problem, we couple multiple tasks via a sparse, directed regularization graph, that enforces each task parameter to be reconstructed as a sparse combination of other tasks selected based on the task-wise loss. We present two different algorithms that jointly learn the task predictors as well as the regularization graph. The first algorithm solves for the original learning objective using alternative optimization, and the second algorithm solves an approximation of it using curriculum learning strategy, that learns one task at a time. We perform experiments on multiple datasets for classification and regression, on which we obtain significant improvements in performance over the single task learning and existing multitask learning models.

1. Introduction

Multi-task learning (Argyriou et al., 2008; Caruana, 1997; Kang et al., 2011; Kumar & Daume III, 2012) aims to improve the generalization ability of the learners for different tasks by jointly training them. While multi-task learning has shown to outperform single-task learning in most cases, the performance gain is usually small since not all tasks benefit from the joint learning, which makes some of the participating learners to suffer from performance degeneration. This problem is known as 'negative transfer', and is caused by the main assumption in most MTL methods, which assumes that the information transfer will be symmetric between any two participating tasks that are coupled by the joint learning; that is, the information transfer from task *a* to *b* will be the same as the information transfer from task *b* to *a*, if they are related. However, this symmetry assumption will not be always beneficial for joint learning, since some tasks will be easier implying lower training task-specific loss while some others will be more difficult with higher training loss. For example, suppose that the task *a* is to predict whether a visual instance 'has wheels' or not, and task *b* is to predict if a given visual object 'is fast'. Obviously the former will be much easier since it can be easily predicted from the visual features, while the predicting the latter from visual information is not straightforward. This resulting in learning a more confident predictor for task *a*, and we would only want to regularize the learning of task *b* with the task *a*, although the two tasks are related. Still, a conventional multitask learning method will consider the two tasks as equal, which will result in the former classifier suffering from negative transfer.

To overcome this intrinsic limitation of conventional symmetric multi-task learning methods, we need to allow for asymmetric information transfer between the tasks, such that the amount of information transfer from a confident predictor to a less confident one is larger than the other way around. We can achieve this objective by learning a weighted directed regularization graph between the tasks, such that more confident learners regularize the learning of less confident ones, but not vice versa. Further, the graph should be sparse since we do not want transfer to happen between unrelated tasks. Based on these two ideas, we propose Asymmetric Multi-task Learning (AMTL), which simultaneously learns such sparse directed regularization graph along with the predictors for each task. To this end,

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48. Copyright 2016 by the author(s).

we first describe the generic asymmetric multitask learning problem, and provide formal analysis on it. Then, we propose two algorithms to solve the problem, where the first algorithm simultaneously trains the learners and the regularization graph via alternating optimization, while the second algorithm solves the problem in the curriculum learning fashion, which iteratively add in learners in the order of increasing task loss and the similarities to later tasks.

We validate our method on multiple datasets for classification and regression tasks, and obtain significant improvement over the single-task learning and exiting multi-task methods. We also show that the performance improvement comes from the suppression of negative transfer, and AMTL obtains even greater performance on datasets with large imbalance in the number of training instances across tasks, which is quite common in real-world data. Finally, we show that AMTL can be also used as an analysis tool, since it generates a sparse directed graph between the tasks which allows easy interpretation of the relations among the tasks.

Our contributions in this work are threefold:

- We tackle the novel problem of learning asymmetric task relations in a multi-task learning framework, to avoid the well-known negative transfer problem.
- We propose a novel multi-task learning formulation, that allows to jointly learn a regularization graph along with the task predictors, and propose both an alternating optimization algorithm and a curriculum learning algorithm to solve the problem.
- We provide a theoretical justification of our AMTL formulation, and empirically validate that it outperforms the symmetric multitask learning methods from reduced negative transfer, with experiments on multiple datasets.

2. Related Work

Transfer learning Our regularization term closely resembles the least square-based regularization term in Tommasi et al. (2010), which is used to perform multi-model transfer for a least-square one-vs-all SVM. However, our method is generic for any loss and learns sparse transfer weight to transfer only from relevant categories to avoid negative transfer, while their method is tied to the specific LS-SVM formulation and learns non-sparse weight due to having a ℓ_2 norm regularization. Further, our method considers optimal selection of tasks and regularization graph to avoid negative transfer in multi-task learning setting, while they assume a fixed set of source and target tasks.

Multitask learning Our method is an instance of multitask learning (Caruana, 1997), with the specific focus on avoiding negative transfer, where the transfer between the tasks results in performance degeneration. Most multi-task learning methods either promote sharing of the features or parameters to exploit shared information across multiple tasks. Many recent work are closely related to the feature learning method by Argyriou et al. (2008), which utilizes $(2, 1)$-norm regularization to learn features shared across multiple categories, and solves it using an equivalent convex formulation. However this method enforces sharing also between tasks that are distantly related, which results in negative transfer. To overcome this problem, Kang et al. (2011) propose to enforce feature sharing only between closely related tasks, by alternating between the learning of task groups using integer programming, and perform multitask feature learning for each group. Kumar & Daume III (2012) and Maurer et al. (2012) allow for more flexible sharing between tasks by learning latent parameter bases that are shared across all tasks. Saha et al. (2011) present an online multi-task learning method that is relevant to our work, where the pairwise relationships between tasks are modeled using a positive definite matrix. However, it cannot learn asymmetric task relations since the positive definiteness constraint on the formulation requires the relations to be symmetric. None of the aforementioned multitask learning methods consider the differences across task losses and thus are susceptible to negative transfer. Asymmetric multitask learning was mentioned in some prior literatures, but asymmetry there simply means either having a pre-defined set of main and auxiliary tasks (Leen et al., 2012) or training models for new tasks along with the existing models (Zhang & Yeung, 2010). On the other hand, our method *learns* asymmetric task relationships based on the task loss and relatedness.

Curriculum learning The term ‘curriculum learning’ was first used in Bengio et al. (2009), which showed that learning easier instances first, and then gradually introducing more difficult examples speeds up the training process and finds a good local minimum for non-convex learning. This idea is further developed in Kumar et al. (2010), which measures the ‘easiness’ of a task by examining the margin of each instance to the classifier, and allows active selection of samples by the learner. Lee & Grauman (2011) exploit a similar idea for unsupervised learning, where they iteratively discovered categories while using earlier learned categories as the context of later learned ones. Ruvolo & Eaton (2013) propose a curriculum learning method based on an active task selection, which aims to maximize the task diversity to learn better approximating latent parameter bases shared across all tasks. Perhaps the most relevant work to ours in the context of curriculum learning is Pentina et al. (2015), which regularizes the parameter of a newly learned task to be similar to its immediate predecessor, using ℓ_2 norm regularization. We also use regularizations on the task parameters to transfer knowledge, but we allow to transfer from all previous tasks, while selecting confident tasks to transfer from, based on their loss. Further when selecting a new task, we also consider its similarity to all future tasks, to maximize the expected amount of knowledge transfer across all tasks.

3. Asymmetric Multi-task Learning

We assume that we have T tasks with various degrees of difficulties: for each task $t \in \{1, \ldots, T\}$, we are given training data $\mathcal{D}_t$ that consists of n_t training points. Given $\{\mathcal{D}_t\}_{t=1}^T$ training examples, let $\mathcal{L}(w_t; \mathcal{D}_t)$ be the loss function of model parameter w_t on the training data $\mathcal{D}_t$. For simplicity, we assume that all tasks share the common data space and the model space, and moreover T tasks are positively correlated. Having $\{w_t^*\}_{t=1}^T$ to denote the set of target model parameters, the simplest statistical assumption to this end is that each underlying model parameter is *succinctly* represented by the linear combination of other parameters: for all $t \in \{1, \ldots, T\}$, $w_t^* \approx \sum_{s=1}^T B_{st}^* w_s^*$ where B^* is a $T \times T$ *asymmetric* matrix representing the amounts of information transfer between participating tasks: B_{st}^* is defined as the positive weight of basis w_s^* in representing w_t^*, and the weight vector $\{B_{st}^*\}_{t=1}^T$ is *sparse*. Note that for notational simplicity we can use B_{tt}^*, but it is uniformly defined as 0 for all $t \in \{1, \ldots, T\}$.

Now, we propose a novel learning approach that allows asymmetric information transfer from easier tasks to difficult ones. Here assuming that the reconstruction error $w_t^* - \sum_{s=1}^T B_{st} w_s^*$ follows Gaussian distribution with small variance, we use a regularizer of $\|w_t - \sum_{s=1}^T B_{st} w_s\|_2^2$ for all tasks. To this end, our learning method jointly learns $\{w_t\}_{t=1}^T$ and B from the following optimization problem:

$$\underset{W, B \geq 0}{\text{minimize}} \sum_{t=1}^T \Big\{ \big(1 + \mu \|b_t^o\|_1\big) \mathcal{L}\big(w_t; \mathcal{D}_t\big) + \lambda \Big\| w_t - \sum_{s=1}^T B_{st}\, w_s \Big\|_2^2 \Big\} \qquad (1)$$

where $b_t^o \in \mathbb{R}^{T-1}$ is a vector indicating the amounts of outgoing transfers from task t to all other tasks: $(B_{t1}, \ldots, B_{t(t-1)}, B_{t(t+1)}, \ldots, B_{tT})^\top$, and (λ, μ) are the tuning parameters that decide the relative importances between different terms. Note that $B \geq 0$ represents the set of all element-wise positivity constraints of a matrix, which is an additional constraint just under the assumption that $B^* \geq 0$. Also note that W contains w_t, that is, $W := (w_1, w_2, \ldots, w_T)$, for notational simplicity. One of the important ingredient in (1) is the use of the ℓ_1 norm of b_t^o to encourage sparsity on B, which allows each predictor to be succinctly represented by other predictors.

It is instructive to consider two extreme learning strategies from (1): (i) when the regularization parameter $\lambda = 0$, the minimum can be achieved at $B = 0$, and each task is decomposable and trained independently, (ii) when λ approaches to ∞, on the other hand, the regularization term becomes dominant and we should have nonzero (and possibly dense even with $\|b_t^o\|_1$ term) B_{st} so that the predictors are linearly dependent. Then, the natural question arises on how the pairwise edge B_{st} emerges as λ increases. The following theorem can precisely answer on this:

Theorem 1 *Consider the optimization problem* (1). *Then,* **any** *local optimum* $(\widehat{w}_1, \ldots, \widehat{w}_T, \widehat{b}_1^o, \ldots, \widehat{b}_T^o)$ *of* (1) *satisfies the following statement. For any* $t, u \in \{1, \ldots, T\}$ *such that* $\|\widehat{b}_t^o\|_1 > \rho \|\widehat{b}_u^o\|_1$*, either one of the following conditions is true:*

(a) $\mathcal{L}(\widehat{w}_t; \mathcal{D}_t) \leq \mathcal{L}(\widehat{w}_u; \mathcal{D}_u)$*, or*

(b) For **any** *vector* b_t^o *and* b_u^o *such that* $\|b_t^o\|_1 = \|\widehat{b}_u^o\|_1$ *and* $\|b_u^o\|_1 = \|\widehat{b}_t^o\|_1$*, we have*
$\mathcal{R}\big(\widehat{w}_1, \ldots, \widehat{w}_T, \widehat{b}_1^o, \ldots, \widehat{b}_t^o, \ldots, \widehat{b}_u^o, \ldots, \widehat{b}_T^o\big)$
$\leq \mathcal{R}\big(\widehat{w}_1, \ldots, \widehat{w}_T, \widehat{b}_1^o, \ldots, b_t^o, \ldots, b_u^o, \ldots, \widehat{b}_T^o\big)$

where $\mathcal{R}(W, B)$ *is a regularization term in* (1) *computed at* W *and* B.

Proof sketch. First note that (1) is differentiable and biconvex in $\{w_t\}_{t=1}^T$ and B. Let f be a continuous biconvex function on W and B. Then, for a continuous biconvex function f, it is known that arbitrary stationary point $(\widehat{W}, \widehat{B})$ with zero partial gradient is a partial optimal, meaning that $f(\widehat{W}, \widehat{B}) \leq f(W, \widehat{B})$ for all W as well as $f(\widehat{W}, \widehat{B}) \leq f(\widehat{W}, B)$ for all B (Gorski et al., 2007). Now, suppose that both the conditions (a) and (b) are violated at the same time. Then, there exists some $\bar{b}_t^o$ and $\bar{b}_u^o$ such that $\|\bar{b}_t^o\|_1 = \|\widehat{b}_u^o\|_1$ and $\|\bar{b}_u^o\|_1 = \|\widehat{b}_t^o\|_1$, and $\mathcal{R}\big(\widehat{w}_1, \ldots, \widehat{w}_T, \widehat{b}_1^o, \ldots, \widehat{b}_t^o, \ldots, \widehat{b}_u^o, \ldots, \widehat{b}_T^o\big) > \mathcal{R}\big(\widehat{w}_1, \ldots, \widehat{w}_T, \widehat{b}_1^o, \ldots, \bar{b}_t^o, \ldots, \bar{b}_u^o, \ldots, \widehat{b}_T^o\big)$. Moreover, since $\mathcal{L}(\widehat{w}_t; \mathcal{D}_t) > \mathcal{L}(\widehat{w}_u; \mathcal{D}_u)$, we are able to have better objective with $\bar{b}_t^o$ and $\bar{b}_u^o$ than that with $\widehat{b}_t^o$ and $\widehat{b}_u^o$, fixing all other parameters, which contradicts with the previous theorem on the partial optimal in Gorski et al. (2007). □

Problem (1) is not convex in general, and hence the gradient-based algorithms will find the local optima of (1). Nevertheless, since the problem is biconvex, this theorem tells us that in **any** local optimum, the task with smaller loss will have larger amounts of transfer as long as it does not hurt the structural constraint in the second term of (1).

Discussions on the problem (1) The motivation behind the optimization form (1) is to encourage a heterogeneous knowledge transfer from easy tasks to difficult ones, as shown in Theorem 1 (a). One caveat on this form is the possibility of knowledge transfers from an overfitted task with a small sample size n_t. In order to alleviate this issue, we can define different (or reweighed) loss functions across tasks: new loss function for task t is $\mathcal{L}'(w_t; \mathcal{D}_t) := c_t \mathcal{L}(w_t; \mathcal{D}_t)$ where c_t is a scalar that should depend on the degree of overfitting of task t. When the estimation error is bounded by $\sqrt{1/n}$ for size n training data (this holds for linear or logistic regressions. See Negahban et al. (2012) and the references therein), the natural selection would be $c_t = 1/\sqrt{n_t}$. Practically, this value can be selected on a separate validation set.

Another caveat is that the regularization term could be also reduced by enforcing each task parameter to be represented as a combination of parameters from unrelated confident tasks. However, this does not happen in practice since it will result in increased loss, and we have sparsity regularization on B to select only strongly related tasks to regularize each learner. Specifically, in Algorithm 1, we first train independent task predictors and then learn regularization graphs between them; thus the connection from the unrelated tasks will be dropped in the first iteration, further preventing such negative effect.

Optimization The optimization problem (1) is not convex because of the first term which is the multiplication of functions on w_t and b_t^o. Nevertheless, the optimization problem is convex in B and convex in W as long as the loss function $\mathcal{L}(\cdot)$ is convex in W. For this biconvex optimization problem, in this paper, we utilize an iterative alternating optimization technique that solves convex optimization problem in B or W at a time fixing the other.

In order to solve (1) in W, we fix B and solve the following optimization problem on W:

$$W \leftarrow \underset{W}{\text{argmin}} \sum_{t=1}^{T} \Big\{ \big(1 + \mu \|b_t^o\|_1\big) \mathcal{L}\big(w_t; \mathcal{D}_t\big) + \lambda \Big\| w_t - \sum_{s \neq t} B_{st}\, w_s \Big\|_2^2 \Big\} \qquad (2)$$

which can be handled by the simple gradient descent method in W. Alternatively, (2) can be solved by the block-coordinate based methods: for each task $t \in \{1, \dots, T\}$, we fix $\{w_s\}_{s \in \{1,\dots,T\}\setminus t}$ along with B, and iteratively compute the following optimization problem only on w_t: $w_t \leftarrow \text{argmin}_{w_t} (1 + \mu \|b_t^o\|_1)\, \mathcal{L}(w_t; \mathcal{D}_t) + \lambda \sum_{s=1}^{T} \|w_s - \sum_{u \neq s} B_{us}\, w_u\|_2^2$ where the problem consists of (re-weighed) the loss function and the sum of T ℓ_2-based regularizers.

Algorithm 1 AMTL using Alternating Optimization

```
Input: $\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_T$
  Initialize $B = 0$
  while Predefined stopping criterion is satisfied do
    Fixing $B$, solve (2) and update $W$, whose $t$-th column is $w_t$
    for all $t \in \{1, \dots, T\}$ do
      Fixing $W$ and $\{b_u^i\}_{u \in \{1,\dots,T\}\setminus t}$, solve (3) and update $b_t^i$
    end for
  end while
```

Contrast to the partial optimization problem in W, the problem in the edge weights B is decomposable with respect to the columns of B. Let $b_t^i := (B_{1t}, B_{2t}, \dots, B_{Tt})^\top \in \mathbb{R}^T$ be the t-th column of B for incoming edges to task t. Then, the problem (1) can be rewritten as follows for easier interpretation on B:

$$\underset{W, B \geq 0}{\text{minimize}} \sum_{t=1}^{T} \Big\{ \mu \|\Lambda\, b_t^i\|_1 + \mathcal{L}\big(w_t; \mathcal{D}_t\big) + \lambda \big\| w_t - W\, b_t^i \big\|_2^2 \Big\}$$

where Λ is a $T \times T$ diagonal matrix whose t-th diagonal entry is the loss for task t, $\mathcal{L}\big(w_t; \mathcal{D}_t\big)$, so that $\sum_{t=1}^{T} \|b_t^o\|_1 \mathcal{L}\big(w_t; \mathcal{D}_t\big) = \sum_{t=1}^{T} \|\Lambda\, b_t^i\|_1$ provided that the loss function is always non-negative. Also note that the equality $\sum_{s \neq t} B_{st}\, w_s = W\, b_t^i$ holds since B_{tt} is fixed as 0. Then, for each task $t \in \{1, \dots, T\}$, we fix W and $\{b_s^i\}_{s \in \{1,\dots,T\}\setminus t}$, and solve

$$b_t^i \leftarrow \underset{b_t^i \geq 0}{\text{argmin}}\ \mu \|\Lambda\, b_t^i\|_1 + \lambda \big\| w_t - W\, b_t^i \big\|_2^2 \qquad (3)$$

which is a weighed LASSO problem with non-negativity constraints. This can be solved by standard optimization techniques such as a proximal gradient descent, and in our implementation we use the weighted lasso implementation in Mairal et al. (2010). Detailed algorithm for our asymmetric multi-task learning is described in Algorithm 1.

AMTL with Curriculum Learning of Multiple tasks We also formulate the curriculum learning of multiple tasks leveraging the paradigm in (1). In the curriculum learning, we find the best order of tasks to be learned when each task is considered based on the knowledge of *all* previously trained tasks. Thus, it can be more efficient than the alternating AMTL algorithm[1], as it requires to train each w_t and b_t^o only once in the entire training process. Let $\mathcal{S}$ denotes the permutation space over T elements, and for some $\pi \in \mathcal{S}$, $\pi(i)$ is the i-th element in permutation π. Then, our goal is now to solve the following optimization problem:

$$\underset{\pi \in \mathcal{S}, W, B \geq 0}{\text{minimize}} \sum_{i=1}^{T} \Big\{ \big(1 + \mu \|b_{\pi(i)}^o\|_1\big) \mathcal{L}\big(w_{\pi(i)}; \mathcal{D}_{\pi(i)}\big) + \lambda \Big\| w_{\pi(i)} - \sum_{j=1}^{i-1} B_{\pi(j)\pi(i)}\, w_{\pi(j)} \Big\|_2^2 \Big\}. \qquad (4)$$

To be clear, it is instructive to contrast this problem against the objective in (1). In (1), every model parameter w_t is assumed to be succinctly reconstructed by *all* other model

[1] This might not be always true since the alternating AMTL algorithm can use parallelism on the matrix computation to speed up the learning process.

parameters. In the curriculum learning setting (4), on the other hand, we have some ordering π in learning processing and each model parameter is supposed to be represented only by the predecessors with respect to π.

Since the problem (4) is a computationally intractable combinatorial problem, we propose a greedy algorithm as a heuristic of solving (4); at every iteration i, we select a task $\pi(i)$ to be learned based on the previous selections $\pi(1), \ldots, \pi(i-1)$ so far. One trivial strategy on selecting a task is to naively decompose the objective of (4) with respect to task t and find the task for i-th iteration from the set of unselected tasks with the minimum value of $(1 + \mu\|b_t^o\|_1)\,\mathcal{L}(w_t; \mathcal{D}_t) + \lambda\|w_t - \sum_{j=1}^{i-1} B_{\pi(j)t}\, w_{\pi(j)}\|_2^2$, given the selections up to $\pi(i-1)$.

It might not be advisable, however, to select a task simply based on the previous selections mainly because it lacks exploration for new types of tasks. Moreover, if some task is already reconstructed well by the previous tasks, then it is not necessary to learn that task at this time, since it could still be learned well at later steps.

Considering this, we propose to select a task that is able to improve the future learning process the most. Suppose that we maintain the set $\mathcal{T} := \{\pi(1), \ldots, \pi(i-1)\}$ for trained tasks and $\mathcal{U} = \{1, \ldots, T\} \backslash \mathcal{T}$ for untrained tasks at each task selection process. We then find a task $t \in \mathcal{U}$ to be learned next as follows:

$$\begin{aligned}(t, b_t^o) \leftarrow \underset{t \in \mathcal{U}, b_t^o}{\operatorname{argmin}} \; & \Big\{ \big(1 + \mu\|b_t^o\|_1\big)\,\mathcal{L}\big(w_t; \mathcal{D}_t\big) \\ & + \lambda \sum_{s \in \mathcal{U} \backslash t} \Big\| w_s - \sum_{j=1}^{i-1} B_{\pi(j)s}\, w_{\pi(j)} - B_{ts} w_t \Big\|_2^2 \Big\}. \end{aligned} \tag{5}$$

Note that the second term above is the sum of all regularizations for untrained tasks *after* training task t. Since all w_t for $t \in \mathcal{U}$ are not trained yet, we initialize them with the single-task learning to measure the similarities to all future tasks, assuming that the predictors from single-task learning are good approximations.

Once we have selected the task to be learned at step i, we solve the following problem to greedily minimize (4):

$$\begin{aligned} w_t \leftarrow \underset{w_t}{\operatorname{argmin}} \; & \Big\{ \big(1 + \mu\|b_t^o\|_1\big)\,\mathcal{L}\big(w_t; \mathcal{D}_t\big) \\ & + \lambda \Big\| w_t - \sum_{j=1}^{i-1} B_{\pi(j)t}\, w_{\pi(j)} \Big\|_2^2 \Big\} \end{aligned} \tag{6}$$

where all edge weights $B_{\pi(j)t}$ from the previous tasks to the current selection t were set in the selection process of (5) at time j. The details of our AMTL with the setting of curriculum learning is described in Algorithm 2.

Algorithm 2 AMTL with Curriculum Learning

```
Input: $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_T$
  Initialize with STL: $W \leftarrow W_{\text{STL}}, \mathcal{U} \leftarrow \{1, 2, \ldots, T\}, \mathcal{T} = \phi$
  for $i = 1$ to $T$ do
    Given $W$ and $\pi(1), \ldots, \pi(i-1)$, find task $t \in \mathcal{U}$ (and $b_t^o$)
    from (5)
    $\pi(i) \leftarrow t$
    Given $B$ and $W \backslash w_t$, set $w_t$ from (6)
    $\mathcal{U} \leftarrow \mathcal{U} \backslash t$
    $\mathcal{T} \leftarrow \mathcal{T} \cup \{t\}$
  end for
```

Loss Functions While $\mathcal{L}(w_t; \mathcal{D}_t)$ in the previous sections is any generic loss , we specifically consider the two popular instances in our experiments. Suppose that each $\mathcal{D}_t$ consists of n_t samples of $\{(x_i, y_i)\}_{i=1}^{n_t}$ where $x \in \mathbb{R}^d$ is a feature vector in the uniform feature space across all tasks and and y is the target response that we predict.

Then, for a classification task where the response space for y is binary $\{0, 1\}$, we use the logistic regression model where the loss function is defined as $\mathcal{L}(w_t; x_t, y_t) := \frac{1}{n_t} \sum_{i=1}^{n_t} \Big[(1 - y_i)\langle x_i,\, w_t\rangle + \log\Big(1 + \exp(-\langle x_i,\, w_t\rangle)\Big)\Big]$. For a regression task where we predict real-valued response y, we consider the squared loss function: $\mathcal{L}(w_t; x_t, y_t) := \frac{1}{n_t}\|y_t - X_t w_t\|_2^2$ where X_t is $n_t \times d$ design matrix on x_t.

4. Experiments

We evaluate the two variants of our methods on multiple datasets for classification and regression, against relevant baselines.

Baselines. We now describe relevant baselines and the two variants of our method.

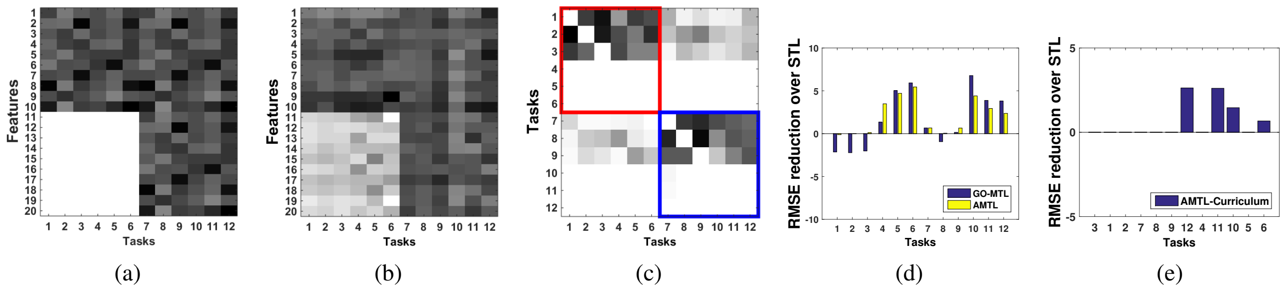
1) STL: Single-task learning method, where each task is learned independently from all the others.

2) MTFL: Multi-task feature learning by Argyriou et al. (2008), which enforces to share features across all tasks. This multitask learning method provides no means for avoiding negative transfer.

3) GO-MTL and SC-MTL: Multi-task learning method that allows grouping and overlap in Kumar & Daume III (2012), which represents each task as a sparse combination of latent parameter bases shared across all tasks. SC-MTL (Maurer et al., 2012) shares the same principle under the dictionary learning framework. These methods account for negative transfer to some degree by allowing selective sharing of information, but are still prone to negative transfer since they do not consider the task loss.

4) Curriculum-simple: Our implementation of the curriculum learning method in Pentina et al. (2015), which allows the regularization to happen only between a task and its direct predecessor, and selects the task without considering the similarity to future tasks.

5) SMTL: A symmetric MTL baseline that implements



(a) (b) (c) (d) (e)

Figure 1. Experimental results on the synthetic dataset. (a) The parameters used to make synthetic data. (b) The model parameters learned by AMTL. (c) Visualization of the regularization graph learned by AMTL. (d) Per-task performance improvement over the STL. The rows denote outgoing edge weights and the columns denote incoming edge weights for each task. (e) Per-task improvement in the selection order by AMTL-curriculum.

(1), with additional requirement that the weight matrix B should be positive semidefinite.

6) AMTL-noLoss: AMTL that does not scale the loss based on the outgoing edge weights. This model will thus consider task relatedness, but not task loss.

7) AMTL: Our asymmetric multi-task learning method, which learns the regularization graph and the task predictor in alternative fashion.

8) AMTL-Curriculum: Our asymmetric multi-task learning method solved by curriculum learning.

For the regularization parameters λ and μ, we find them through cross-validation on designated validation sets.

4.1. Synthetic dataset

We first experiment with a synthetic dataset to validate our model. We generate a synthetic dataset for regression that consists of two groups, where tasks form distinct groups based on the task parameters (See Figure 1(a)). The first group consists of task 1 through 6, and the second group consists of task 7 through 12. In each group, there are 6 tasks whose true parameters are generated from Gaussian distributions with the same mean but different variances. Specifically, the noise level for the first three tasks (task 1-3, 7-9) is set to be low ($\sigma = 5$) while that of remaining tasks is set to be high ($\sigma = 25$) to make tasks in this group more difficult. We generate total of 90 samples per task and use $\{30{,}30{,}30\}$ split for training/validation/test.

From this experiment, we want to show that our AMTL learns transfer weights that consider task relatedness and individual losses, and this helps improve prediction performance, as well as prevent negative transfer. Figure 1(c) shows the transfer weights learned by AMTL, where the columns are weights for the outgoing edges and rows are the weights for the incoming edges. We observe that the relatively easy tasks with low noise (task 1-3 and 7-9) have outgoing edges to tasks with high noise (task 4-6 and 10-12), but not vice versa.

To see if such asymmetric learning of transfer weights actually results in the performance improvement, we also report quantitative evaluation of prediction performance in Table 1. Our methods outperform all baselines, including our own baselines such as SMTL that employs similar formulations but learns symmetric weights, or AMTL-noLoss that does not consider task loss. For further analysis, we examine AMTL's per-task RMSE reduction over STL, against that of GO-MTL (Figure 1(d)). The result clearly shows that the performance improvement achieved by AMTL mostly comes from the suppression of negative transfer. While GO-MTL outperforms AMTL on some tasks (task 5, 6, 10, 11, 12), it also degenerates performance by great degree on several tasks (task 1,2,3,8). On the other hand, AMTL does not result in large accuracy degeneration in any of the tasks.

For analysis on AMTL-curriculum, we show the task selection order and improvement on each task in Figure 1(e). We see that easier tasks in each group (task 1-3, 7-9) are selected at the earlier stage and more difficult tasks (task 4-6, 10-12) are selected in the second half. We observe that while there is no improvement for the earlier selected tasks, there is no degeneration of performance from negative transfer either, and there are significant improvements on later selected, more difficult tasks.

We then compare the runtime of the two AMTL variants on the synthetic dataset. AMTL-curriculum runs significantly faster than AMTL, taking only 7.71 ± 0.16 seconds to run while AMTL takes 167.29 ± 7.41 seconds. Thus, when efficiency is a major concern, AMTL-curriculum might be a better option over AMTL.

4.2. Real dataset

Datasets. For performance evaluation, we use three datasets for classification, and one dataset for regression.

1) MNIST Digits data: This dataset contains $60,000$ training images and $10,000$ test images from 10 handwritten

	Synthetic	MNIST	USPS	School	AWA
STL	20.87±0.36	14.76 ± 0.62	12.44 ± 0.62	10.34±0.13	58.33±1.10
MTFL (Argyriou et al., 2008)	19.34±0.37	14.12±0.55	12.29±0.67	9.92±0.04	65.00±0.42
GO-MTL (Kumar & Daume III, 2012)	19.18±0.39	14.44±1.34	11.92±1.48	**9.87**±0.09	62.46±0.25
SC-MTL (Maurer et al., 2012)	19.85±0.30	14.64±0.50	12.36±0.74	10.12±0.05	66.39±0.65
Curriculum-simple (Pentina et al., 2015)	20.87±0.36	14.56±0.45	12.28±0.15	10.30±0.13	57.37±1.66
SMTL	20.87±0.36	14.00±0.41	12.24±0.58	10.19±0.05	59.05±0.40
AMTL-noLoss	20.86±0.36	14.24±0.53	12.56±0.59	10.18±0.05	58.20±0.32
AMTL	**18.80**±0.22	**12.92** ± 1.37	**11.48** ± 1.21	10.13±0.08	**56.83** ± 1.11
AMTL-Curriculum	20.33±0.29	13.68±1.42	11.84±1.46	10.16±0.05	56.99±1.09

Table 1. Task performance on multiple datasets. We report the root mean squared error for regression, and the mean classification error (%) for classification, as well as the standard error for 95% confidence interval over 5 splits.

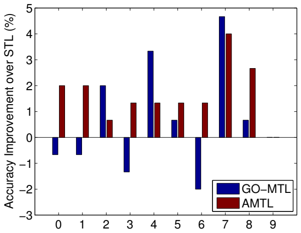


Figure 2. Per-class performance improvement of GO-MTL and AMTL, on the MNIST dataset.

digits (0-9). The raw image has 28 × 28 dimensions, and we reduce the dimensionality to 64 using PCA. We use 5 random splits for training/validation/test datasets with 1000/500/500 instances, following the procedure of Kang et al. (2011) and Kumar & Daume III (2012)

2) USPS Digits data: Another handwritten digit dataset, that is composed of $7,291$ training images and $2,007$ test images. The raw images are 16×16 grayscale images, and we reduce the dimensionality to 87 using PCA. We generate 5 random splits by selecting 1000 random samples from the training set, and select two sets of 500 random images from the test set, to be used as validation and test.

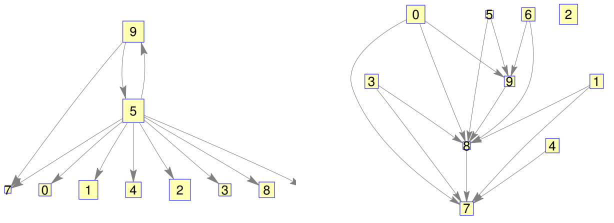
3) School dataset: This regression dataset consists of exam scores of 15,362 students from 139 schools, where the scores are real values. There are 28 features per each instance, but we use 26 binary-valued features following the procedure of Argyriou et al. (2008). We use the first 5 splits among 10 splits provided by Argyriou et al. (2008).

4) AWA: This dataset (Lampert et al., 2009) contains $30,475$ images from 50 animal classes, such as *chimpanzee*, *giant panda*, *leopard*, *persian cat*, *hippopotamus*. For training/validation/test splits, we used 30/30/30 images. For features, we use the provided 4096-D DeCAF features, and reduced their dimensionality to 500.

Table 1 shows the prediction performance of the baselines and our methods on all four datasets. Both AMTL and AMTL-Curriculum outperform the STL baseline, as well as the baseline multi-task / curriculum learning methods, except for the School dataset where GO-MTL performs better. We attribute this to the fact that the different tasks in the school dataset are essentially targeting the same problem on different data. Curriculum-simple obtains small performance improvements on all datasets, which could be due to the weaker regularizer that only uses the previously selected task for knowledge transfer. On all datasets, AMTL performs slightly better than AMTL-Curriculum. The reason for this could be that the AMTL-Curriculum requires to have strict directional graph even between two tasks with similar loss, based on the order they are selected. However, AMTL-Curriculum still has advantages in terms of efficiency. To show that our performance improvement is coming from the reduction in the negative transfer, we plot the per-task accuracy improvements over the STL of GO-MTL, and AMTL on the MNIST dataset (Figure 2). Our method, as expected, improves the prediction performance on all tasks. GO-MTL, on the other hand, does well on some, even outperforming ours (task 2, 4, 7) but at the same time degenerates performance on several other tasks, and thus results in lower overall performance compared to ours. This per-class result agrees with the result on the synthetic dataset, and suggests that our AMTL could be even more useful for cases where losing performance on any of the tasks is undesirable.

Our MTL variants with similar formulation but with symmetric weights (SMTL), or no weights on the loss (AMTL-noLoss) do not work as well as the AMTL, or AMTL-Curriculum, which suggests that allowing asymmetric transfer and deciding the transfer direction on the task loss is essential to the success of our model.

For qualitative analysis, we visualize the learned regularization graph for AMTL and AMTL-Curriculum on the MNIST dataset in Figure 3(a) and (b), varying the size of the nodes based on the task loss (accuracy). We observe that the edges come from the tasks with low loss and go into the ones with higher loss. Further, the tasks that are associated through the regularization graph are mostly intuitively relevant ones. For example, in Figure 3(a), 5 and 9 are coupled and there is a strong edge from 9 to 7, and



(a) AMTL (b) AMTL-Curriculum

Figure 3. Regularization graph learned on the MNIST dataset.

in Figure 3(b), 0 and 6 are selected to regularize the learning of 9, which makes sense when considering their shapes. Note that the regularization graph from AMTL-Curriculum generates a non-cyclic directed graph while AMTL allows such cyclic dependencies.

4.3. Experiment on datasets with large variance in number of training instances

While most standard multi-task learning datasets have similar number of training instances per task, this might not hold in real-world scenarios where we might have largely uneven number of instances per task. For example, in ImageNet dataset (Deng et al., 2009) for object categorization task, some object categories have more than 3000 images per class, while some others have less than 10 instances. This imbalance results from the discrepancy in their actual occurrence frequencies in the real world. As mentioned in Section 3, our model can incorporate the number of training instances to prevent the models overfitted to small number of instances from transferring to models with larger number of instances. As we discussed after introducing (1), we can control the amount of transfers by a task weight c_t, allowing more outgoing transfers from tasks with more samples. We call this variant AMTL-imbalance. We perform additional experiments on two datasets that exhibit such a training size imbalance, to see how this variant can benefit in practice.

1) Synthetic dataset: This is a synthetic dataset created as in the description in Section 4.1, but with different number of training instances per task. We use the same task parameters as the synthetic dataset in 4.1, except that the noise level is set to 50 for all tasks. Here, we used 500 training instances for task 1-3, and 7-9, and 5 instances for task 4-6 and 10-12. We use 50 instances for validation and test.

2) ImageNet-Room: This dataset is a subset of the ImageNet dataset that contains 20 classes under the superclass *room*. In this dataset, some classes have as many as 1,000 instances (*refectory*, *palace*, *salon*), while some classes have as few as 30 instances (*washroom*, *concert hall*, *tollbooth*). For each class, we randomly selected 20 instances for test, and used remaining instances for training. For features, we extracted 4096-D Caffe features, and reduced their dimensionality to 500 using PCA, for faster learning.

Method	Synthetic	ImageNet-Room
STL	54.05±0.82	45.85±1.36
MTFL	55.44±0.94	47.95±1.20
GO-MTL	55.18±0.85	47.05±1.35
SC-MTL	54.37±0.98	47.60±1.26
Curriculum-simple	54.05±0.82	45.25±1.26
SMTL	54.09±0.85	46.00±1.16
AMTL-noLoss	53.93±0.76	48.20±1.33
AMTL	53.19 ±0.77	40.80±1.46
AMTL-imbalanced	**53.09 ± 0.76**	**40.00±1.71**

Table 2. Prediction error of different models on the Imbalanced datasets. We report the average error over 5 random splits, as well as the standard error at 95% confidence interval.

We report the results of this experiment in Table 2. On both datasets, our AMTL models significantly outperforms the STL, especially on the ImageNet-Room dataset, while GO-MTL significantly degenerates performance. The degenerate performance of GO-MTL might be due to the negative transfer caused by enforcing the high-confident models to share bases with low-confident models trained with few training samples. Further, we see that the AMTL-imbalanced works better than AMTL, which shows that considering number of training instances also help with the prevention of negative transfer.

5. Conclusion

In this work, we propose Asymmetric Multitask Learning (AMTL) based on task relatedness and loss, which enables to perform asymmetric transfer of information between tasks in multi-task learning. By allowing to select the regularization parameters based on the source task loss, AMTL can avoid negative transfer from less confident, and difficult tasks to more confident ones. To select few useful relations while preventing transfer between unrelated tasks, we add in non-negativity and sparsity constraints to learn a sparse non-negative regularization graph. We solve this task relation graph learning problem using both an alternative learning algorithm and curriculm learning algorithm, and validate their performances on multiple datasets, on which we obtain significant improvements in the task performance. We further show that our method minimizes the negative transfer by comparing the per-task performance with a symmetric multitask learning baseline, and that it works even better when training data distribution across tasks is largely imbalanced, with additional experiments.

Acknowledgements This work was supported by Samsung Research Funding Center of Samsung Electronics under project number SRFC-IT150203.

References

Argyriou, Andreas, Evgeniou, Theodoros, and Pontil, Massimiliano. Convex Multi-task Feature Learning. *Machine Learning*, 73(3):243–272, 2008.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. Curriculum learning. In *ICML*, 2009.

Caruana, R. Multitask Learning. *Machine Learning*, 1997.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and ei, L. Fei-F˙ Imagenet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

Gorski, Jochen, Pfeuffer, Frank, and Klamroth, Kathrin. Biconvex sets and optimization with biconvex functions: a survey and extensions. *Math. Meth. of OR*, 66(3):373–407, 2007.

Kang, Zhuoliang, Grauman, Kristen, and Sha, Fei. Learning with whom to share in multi-task feature learning. In *ICML*, 2011.

Kumar, Abhishek and Daume III, Hal. Learning task grouping and overlap in multi-task learning. In *ICML*, 2012.

Kumar, M Pawan, Packer, Ben, and Koller, Daphne. Self-Paced Learning for Latent Variable Models. In *NIPS*, pp. 1–9, 2010.

Lampert, Christoph, Nickisch, Hannes, and Harmeling, Stefan. Learning to Detect Unseen Object Classes by Between-Class Attribute Transfer. In *CVPR*, 2009.

Lee, Y. J. and Grauman, K. Learning the easy things first: Self-paced visual category discovery. In *CVPR*, pp. 1721–1728, 2011.

Leen, Gayle, Peltonen, Jaakko, and Kaski, Samuel. Focused multi-task learning in a gaussian process framework. *Machine Learning*, 89(1-2), 2012. ISSN 0885-6125.

Mairal, Julien, Bach, Francis, Ponce, Jean, and Sapiro, Guillermo. Online learning for matrix factorization and sparse coding. *J. Mach. Learn. Res.*, 11:19–60, March 2010. ISSN 1532-4435.

Maurer, Andreas, Pontil, Massimiliano, and Romera-Paredes, Bernardino. Sparse coding for multitask and transfer learning. In *ICML*, 2012.

Negahban, S., Ravikumar, P., Wainwright, M. J., and Yu, B. A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers. *Statistical Science*, 27(4):538–557, 2012.

Pentina, Anastasia, Sharmanska, Viktoriia, and Lampert, Christoph H. Curriculum learning of multiple tasks. In *CVPR*, June 2015.

Ruvolo, Paul and Eaton, Eric. Active task selection for lifelong machine learning. In *AAAI*, July 2013.

Saha, Avishek, Rai, Piyush, III, Hal Daumé, and Venkatasubramanian, Suresh. Online learning of multiple tasks and their relationships. In *AISTATS 2011*, pp. 643–651, 2011.

Tommasi, T., Orabona, F., and Caputo, B. Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In *CVPR*, 2010.

Zhang, Yu and Yeung, Dit-Yan. A convex formulation for learning task relationships in multi-task learning. In *UAI*, pp. 733–442, 2010.