Adaptive Sharing for Image Classification

Li Shen[1], **Gang Sun**[1,2], **Zhouchen Lin**[3,4,*], **Qingming Huang**[1,5,*], **Enhua Wu**[2,6]
[1] University of Chinese Academy of Sciences, Beijing, China
[2] State Key Lab. of Computer Science, Inst. of Software, CAS, Beijing, China
[3] Key Laboratory of Machine Perception (MOE), School of EECS, Peking University, Beijing, China
[4] Cooperative Medianet Innovation Center, Shanghai, China
[5] Key Lab. of Intell. Info. Process., Inst. of Comput. Tech., Chinese Academy of Sciences, China
[6] University of Macau, Macao, China

Abstract

In this paper, we formulate the image classification problem in a multi-task learning framework. We propose a novel method to adaptively share information among tasks (classes). Different from imposing strong assumptions or discovering specific structures, the key insight in our method is to selectively extract and exploit the shared information among classes while capturing respective disparities simultaneously. It is achieved by estimating a composite of two sets of parameters with different regularization. Besides applying it for learning classifiers on pre-computed features, we also integrate the adaptive sharing with deep neural networks, whose discriminative power can be augmented by encoding class relationship. We further develop two strategies for solving the optimization problems in the two scenarios. Empirical results demonstrate that our method can significantly improve the classification performance by transferring knowledge appropriately.

1 Introduction

Image classification is a core problem in computer vision and artificial intelligence. Learning with a number of class labels poses a great challenge on traditional multi-class classification models, where training multiple classifiers is typically independent or mutually exclusive. The task of distinguishing one class from hundreds of class labels would be difficult, especially when its training data is insufficient.

Much of the effort in deploying algorithms is devoted to leveraging rich task relationship to transfer information [Caruana, 1997; Thrun and Pratt, 1998; Evgeniou and Pontil, 2004]. The learning paradigms aim to achieve better generalization performance by encouraging common knowledge to be shared across related ones. Accordingly, the crucial aspect is the introduction of hypothesis to model the relatedness among tasks. For example, widely used assumptions that task parameters (i.e., classifier parameters) either lie in a common feature subspace [Obozinski *et al.*, 2006; Argyriou *et al.*, 2008; Liu *et al.*, 2009] or share a common probabilistic prior [Yu *et al.*, 2005; Fei-Fei *et al.*, 2006], are essentially based on the hypothesis that all tasks are related and all relevant information should be shared. When such strong assumptions do not hold, such sharing may incur adverse effect on overall performance. To avoid this, some methods have been proposed to discover specific structures, such as outliers [Gong *et al.*, 2012; Pu *et al.*, 2014] or disjoint groups [Zhou *et al.*, 2011; Kang *et al.*, 2011; Srivastava and Salakhutdinov, 2013].

Regarding the realistic classification problems, it is complicated to model class relatedness in the target space. Determining how to share is hard to be addressed accordingly. For example, the introduction of disjoint group structure reflects the desire of intra-group sharing, which drives the classifier parameters in a group close to each other. As a matter of fact, the sharing among classes usually forms a continuum in the more realistic setting. Some classes are less related than others even if they are partitioned to a group. On the other hand, it is imperative to highlight the specific features of one class against others even if they have much in common, since the original goal of developing model is to distinguish the set of classes. A robust method is needed to effectively share information and identify individual difference.

In this regard, we propose an Adaptive Sharing method for image classification. A distinct insight from our method is to selectively share information among classes while capturing respective disparities simultaneously. The learning model is expected to leverage feature relevance when it exists, but not require it strictly satisfying certain structure. This goal is achieved by estimating a composite of two parameter sets with different types of regularization. The classifier parameters for all classes are decomposed into two parts: one corresponds to the shared features and the other corresponds to the class-specific features. A nuclear norm penalty is exploited on the first part to capture the underlying relatedness structure among classes and an element-wise sparsity penalty is imposed on the second part to highlight the disparities of each class. The objective is formulated as a non-smooth convex optimization problem when given the feature space. We

*Corresponding author.

adopt the accelerated proximal gradient algorithm [Beck and Teboulle, 2009] to solve the problem.

Besides, we propose to learn the classifier parameters in the context of deep architectures (e.g., convolutional neural networks) by incorporating Adaptive Sharing. Convolutional neural networks with discriminative training have obtained state-of-the-art results on many classification problems. Our method augments the network by encoding the class relatedness. We also develop an effective way to solve the optimization by using stochastic gradient descent algorithm with approximation on low-rank constraint. In summary, the main contributions in this paper are as follows:

- We propose a novel Adaptive Sharing method which selectively shares information among classes and captures the class-specific properties simultaneously. Compared with imposing strong assumptions or discovering specific structures, we provide an elegant way to appropriately transfer knowledge among classes.
- Besides applying it for learning classifiers on pre-computed features, we also integrate Adaptive Sharing with deep neural networks, where the performance can be improved by encoding class relatedness structure. Consequently, we develop different optimization strategies in the two scenarios. Experimental results on multiple challenging datasets demonstrate the efficacy of such selectively transfer for improving the overall classification performance.

2 Related Work

Image classification in real world scenario has drawn increasing attention. Complex appearance variations and class correlation bring in the difficulties for classifying many classes. Much work is proposed to study and exploit the relatedness among classes to transfer information in a multi-task learning paradigm. A family of methods are developed based on sharing a prior in the hierarchical Bayesian framework [Fei-Fei *et al.*, 2006; Archambeau *et al.*, 2011]. Another direction is formulating the approaches in the regularization framework where the tasks are assumed to lie in a common feature subspace [Argyriou *et al.*, 2007; Liu *et al.*, 2009]. However, these methods typically assume strong shared relationship among tasks, which might degrade the overall performance due to the information transfer among unrelated tasks. Some methods are further proposed to discover relatedness structure for sharing. For example, a mixed penalty is adopted in [Mei *et al.*, 2012], outlier tasks are detected in [Gong *et al.*, 2012; Pu *et al.*, 2014] and task grouping is learnt in [Zhou *et al.*, 2011; Kang *et al.*, 2011]. Different from our method, these methods encourage the relatedness satisfying certain structural bracket. Moreover, the work in [Jalali *et al.*, 2010; Chen *et al.*, 2012] captures the inherent relationship among tasks while allowing the existence of different features. The two work provide theoretical analysis based on the linear feature space, whereas we are concerned with a more realistic problem and further incorporate our method with deep architectures.

Alternatively, much effort is devoted to developing feature representation learning [Krizhevsky *et al.*, 2012; Lin *et al.*, 2014; Lee *et al.*, 2014; Stollenga *et al.*, 2014; He *et al.*, 2014; Simonyan and Zisserman, 2014], which achieves state-of-the-art performance on image classification. However, the development of these models does not take advantage of class relatedness. The work [Deng *et al.*, 2014] exploits semantic prior in the deep model, which is different from implicitly learning the relatedness structure in our method. In [Srivastava and Salakhutdinov, 2013], a group-based structure is estimated with deep model iteratively. As any change on class partitioning would lead to the re-training of overall network, the method suffers from considerable time cost for reaching a plateau and might be intractable for dealing with many classes. Our method adopts a more concise and effective way to combine the class relationship.

3 Adaptive Sharing Approach

Assume we have a set of training images $\mathcal{X} = \{x_i\}_{i=1}^N$. $\mathcal{Y} = \{\mathbf{y}_i\}_{i=1}^N$ is the corresponding label set. $\mathbf{y}_i$ is a K dimensional vector (whose value can be binary or one-of-K) for indexing target classes. The learning of each classifier is regarded as a single task. Notation $\|\cdot\|_1$, $\|\cdot\|_F$ and $\|\cdot\|_*$ denote the ℓ_1 norm, Frobenius norm and nuclear norm of matrix [Lin *et al.*, 2011], respectively. The nuclear norm is the sum of singular values.

In this section, we first describe our method based on pre-computed features and the optimization strategy by using accelerated proximal gradient algorithm. Then we incorporate Adaptive Sharing with convolutional neural networks and present the corresponding optimization strategy.

3.1 Learning on pre-computed features

Suppose each image $x \in \mathcal{X}$ has been represented by a pre-computed feature vector $\mathbf{x} \in \mathbb{R}^D$. Let $\mathbf{w}_k \in \mathbb{R}^D$ denote the corresponding parameter vector of task k (i.e., the classifier parameters of class k) which is a column of the parameter matrix $\mathbf{W} \in \mathbb{R}^{D\times K}$. In the standard learning paradigm where each task is learnt independently, the objective function typically takes the form:

$$\min_{\mathbf{W}} \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{N} \ell\left(\mathbf{w}_k^T\mathbf{x}_i,\ y_i^k\right) + \lambda\|\mathbf{W}\|_F^2, \qquad (1)$$

where λ is the regularization factor, and $\ell(\mathbf{w}_k^T\mathbf{x}_i, y_i^k)$ denotes the loss between the prediction $\mathbf{w}_k^T\mathbf{x}_i$ and the true value y_i^k. When considering feature selection [Obozinski *et al.*, 2006], ℓ_1 norm regularization is utilized instead.

In multi-task learning paradigm, the tasks are expected to learn jointly and share a common feature subspace, such as imposing a low-rank structure [Argyriou *et al.*, 2008]:

$$\min_{\mathbf{W}} \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{N} \ell\left(\mathbf{w}_k^T\mathbf{x}_i,\ y_i^k\right) + \lambda\|\mathbf{W}\|_*. \qquad (2)$$

Such a regularizer enforces the strict sharedness, which might drive multiple classifiers too close. However, with respect to classification problems, the essential goal is to distinguish between classes. For example, it is imperative to highlight the features differentiating "Siberian husky" and "Malamute" although they share much common information. Thus,

we suppose that each task depends on the shared information and the additional specific properties.

Here we leverage a composite of two parameter vectors, $\mathbf{c}_k$ and $\mathbf{s}_k$, to represent the classifier parameters towards class k, i.e., $\mathbf{w}_k = \mathbf{c}_k + \mathbf{s}_k$. The two components correspond to the shared features and the specific features, respectively. Let $\mathbf{C} = (\mathbf{c}_1, \cdots, \mathbf{c}_K)$ and $\mathbf{S} = (\mathbf{s}_1, \cdots, \mathbf{s}_K)$. We introduce different structures on the matrics $\mathbf{C}$ and $\mathbf{S}$. A low-rank structure is exploited to capture the inherent relatedness among tasks and an element-wise sparsity structure is leveraged to highlight the disparities of each task simultaneously. Formally, the learning problem can be formulated as:

$$\min_{\mathbf{C},\mathbf{S}} \frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{N} \ell\left((\mathbf{c}_k + \mathbf{s}_k)^T \mathbf{x}_i, y_i^k\right) + \lambda_c \|\mathbf{C}\|_* + \lambda_s \|\mathbf{S}\|_1, \tag{3}$$

where λ_c and λ_s are the penalty factors. The nuclear norm penalty has the effect of encouraging a low rank solution on matrix $\mathbf{C}$ for coupling the related tasks. The ℓ_1 norm penalty is adopted to characterize the specific features by encouraging the sparsity on matrix $\mathbf{S}$.

To solve the optimization problem (3), we use accelerated proximal gradient algorithm [Beck and Teboulle, 2009]. Let $\mathcal{L}(\mathcal{X}, \mathcal{Y}; \mathbf{C}, \mathbf{S})$ symbolically denote the empirical loss:

$$\mathcal{L}(\mathcal{X}, \mathcal{Y}; \mathbf{C}, \mathbf{S}) = \frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{N} \ell\left((\mathbf{c}_k + \mathbf{s}_k)^T \mathbf{x}_i, y_i^k\right), \tag{4}$$

$\mathbf{Z}$ denote the variables to be optimized,

$$\mathbf{Z} = \begin{pmatrix} \mathbf{C} \\ \mathbf{S} \end{pmatrix}, \mathbf{C} \in \mathbb{R}^{D \times K}, \mathbf{S} \in \mathbb{R}^{D \times K}, \tag{5}$$

$g(\cdot)$ and $h(\cdot)$ refer to the smooth term and non-smooth convex term, respectively,

$$g(\mathbf{Z}) = \mathcal{L}(\mathcal{X}, \mathcal{Y}; \mathbf{C}, \mathbf{S}), h(\mathbf{Z}) = \lambda_c \|\mathbf{C}\|_* + \lambda_s \|\mathbf{S}\|_1. \tag{6}$$

The updating at the t-th iteration performs as follows:

$$\mathbf{Z}_{t+1} = \arg\min_{\mathbf{Z}} \left(h(\mathbf{Z}) + \frac{1}{2\eta_t} \|\mathbf{Z} - (\mathbf{U}_t - \eta_t \nabla g(\mathbf{U}_t))\|_F^2 \right), \tag{7}$$

where η_t denotes step size. $g(\mathbf{Z})$ is approximated by a quadratic local model around $\mathbf{U}_t$. The variable $\mathbf{U}_t$ can be set as a combination of $\mathbf{Z}_t$ and $\mathbf{Z}_{t-1}$ from previous iterations:

$$\mathbf{U}_t = \mathbf{Z}_t + \frac{b_{t-1} - 1}{b_t} (\mathbf{Z}_t - \mathbf{Z}_{t-1}), \tag{8}$$

where $b_t = \left(1 + \sqrt{4b_{t-1}^2 + 1}\right)/2$ for $t \geq 1$, and $b_0 = 1$. Considering that (7) takes an equivalent form:

$$\min_{\mathbf{C},\mathbf{S}} \frac{1}{2} \|\mathbf{C} - \hat{\mathbf{C}}_t\|_F^2 + \frac{1}{2} \|\mathbf{S} - \hat{\mathbf{S}}_t\|_F^2 + \hat{\lambda}_c \|\mathbf{C}\|_* + \hat{\lambda}_s \|\mathbf{S}\|_1, \tag{9}$$

where $\begin{pmatrix} \hat{\mathbf{C}}_t \\ \hat{\mathbf{S}}_t \end{pmatrix} \triangleq \hat{\mathbf{Z}}_t = \mathbf{U}_t - \eta_t \nabla g(\mathbf{U}_t)$, and $\hat{\lambda}_c = \eta_t \lambda_c, \hat{\lambda}_s = \eta_t \lambda_s$. We can leverage the decomposability in (9) to optimize variables $\mathbf{C}$ and $\mathbf{S}$ separately, and the closed-form solutions can be obtained [Lin *et al.*, 2011], respectively.

3.2 Integrating Adaptive Sharing with Deep Neural Networks

Deep neural networks have shown strong power on image classification. We aim to integrate Adaptive Sharing into deep neural network framework to augment the network by encoding the relatedness among classes. Our model can be implemented as a standalone layer in such a framework. We exploit it to replace the last full-connected layer in deep neural networks (such as convolutional neural networks [Krizhevsky *et al.*, 2012]), that is to say, the last layer weight parameters (connected to K nodes corresponding to the K classes) are comprised of two components.

The neural network can be regarded as a feature space projection for each image x_i. Different from learning classifiers on pre-computed features (in Section 3.1), the parameters in the projection should be learnt jointly. Let $\boldsymbol{\theta}$ denote the parameters in the network except the ones at the last layer. Then the empirical loss in (4) can be reformulated with negative log-likelihood:

$$\mathcal{L}(\mathcal{X}, \mathcal{Y}; \mathbf{C}, \mathbf{S}, \boldsymbol{\theta}) = -\frac{1}{N} \sum_{i=1}^{N} \log P(\mathbf{y}_i | x_i, \mathbf{C}, \mathbf{S}, \boldsymbol{\theta}). \tag{10}$$

$\boldsymbol{\theta}$ can be regarded as the parameters to generate the representation for image x_i. The matrices $\mathbf{C}$ and $\mathbf{S}$ compose the last layer weight parameters, i.e., $\mathbf{w}_k = \mathbf{c}_k + \mathbf{s}_k$, which denote the weight parameters towards class k.

The optimization can be done using stochastic gradient descent method with mini-batches. However, the problem is non-trivial due to the low-rank constraint on the matrix $\mathbf{C}$. Therefore, we adopt an operator $\Omega_\varepsilon(\mathbf{C})$ [Mei *et al.*, 2012] to approximate the nuclear norm penalty $\lambda_c \|\mathbf{C}\|_*$:

$$\Omega_\varepsilon(\mathbf{C}) = \min_{\mathbf{Q}} \left(\frac{1}{2\varepsilon} \|\mathbf{Q} - \mathbf{C}\|_F^2 + \lambda_c \|\mathbf{Q}\|_* \right), \tag{11}$$

where ε is the approximation factor. The approximation $\Omega_\varepsilon(\mathbf{C})$ is convex and smooth with respect to $\mathbf{C}$. Then the gradient can be computed as:

$$\nabla \Omega_\varepsilon(\mathbf{C}) = \lambda_c (\mathbf{C} - \mathbf{Q}^*), \tag{12}$$

where $\mathbf{Q}^* = \arg\min_{\mathbf{Q}} \left(\frac{1}{2\varepsilon} \|\mathbf{Q} - \mathbf{C}\|_F^2 + \lambda_c \|\mathbf{Q}\|_* \right)$. With respect to nuclear norm, $\mathbf{Q}^*$ can be computed with a closed-form expression by utilizing the soft-thresholding operator on the singular values of the matrix $\mathbf{C}$ [Lin *et al.*, 2011]. Consequently, the gradient of the regularized loss over a batch of data $\mathcal{D} = \{x_i, \mathbf{y}_i\}$ is estimated as:

$$\left\langle \frac{\partial \mathcal{L}(x_i, \mathbf{y}_i)}{\partial \mathbf{C}} \right\rangle_{\mathcal{D}} + \nabla \Omega_\varepsilon(\mathbf{C}), \tag{13}$$

where $\langle \cdot \rangle_{\mathcal{D}}$ denotes the average operator over the batch $\mathcal{D}$. The update rule for the parameters $\mathbf{S}$ and $\boldsymbol{\theta}$ follows the standard algorithm [Jia *et al.*, 2014].

4 Experiments

We present extensive empirical studies to evaluate our Adaptive Sharing learning in two scenarios: pre-computed features and deep neural networks. We first compare our method with

Figure 1: Example images from the CUB-200-2010 dataset.

Table 1: Evaluation on the CUB-200-2010 dataset (Birds200) and a subset of 14 species (Birds14), measured by accuracy (Acc) and mean Average Precision (mAP), respectively.

Method	Birds200 Acc	Birds14 mAP
FGC methods		
Multi-Cue [Khan *et al.*, 2011]	22.4%	–
Birdlets [Farrell *et al.*, 2011]	–	40.3%
TriCoS [Chai *et al.*, 2012]	25.5%	–
KDES [Bo *et al.*, 2010]	26.4%	42.5%
UTL [Yang *et al.*, 2012]	28.2%	–
Random template [Yao *et al.*, 2012]	–	44.7%
Det. + Seg. [Angelova and Zhu, 2013]	30.2%	–
MTL methods		
JFS [Argyriou *et al.*, 2007]	21.7%	38.9%
CMTL [Zhou *et al.*, 2011]	22.0%	40.6%
GMTL [Pu *et al.*, 2014]	28.4%	45.7%
Adaptive Sharing	**31.3%**	**49.1%**

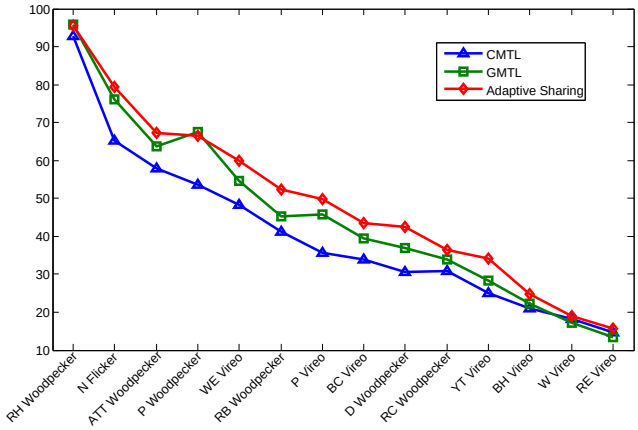

Figure 2: Per-class AP (%) of CMTL, GMTL and Adaptive Sharing on Birds14 (with abbreviated names). The species are sorted by the per-class AP of Adaptive Sharing.

some representative multi-task learning methods in the fine-grained classification problem and assess the performance on a widely used dataset CUB-200-2010 [Welinder *et al.*, 2010]. Moreover, we evaluate it on two challenging datasets, CIFAR-100 [Krizhevsky, 2009] and ImageNet 2012 classification dataset [Russakovsky *et al.*, 2014], in the context of deep architectures.

4.1 Evaluation with Pre-computed Features

The CUB-200-2010 (Birds200) is a widely used dataset for fine-grained classification (FGC). It contains 6,033 images of birds belonging to 200 species, where only 15 images per class are used for training, the rest are used for testing. Some example images from the dataset are shown in Fig. 1. Each image is first cropped against the provided bounding box and resized such that the longer side is no more than 300 pixels. Our method does not specify features, and we use kernel descriptors (KDES) [Bo *et al.*, 2010] as the image-level representation. Specifically, four types of the KDES are applied: gradient-based, color-based, normalized color-based, and local-binary-pattern-based. The patch size is set to 16×16, and the stride is set to 8 pixels. We adopt the squared hinge loss [Yang *et al.*, 2009]. We change λ_c from 0 to 1.0 with step 0.1, and λ_s from 0.001 to 1.0 with ratio 10.

We compare our Adaptive Sharing with three representative multi-task learning methods (MTL): JFS [Argyriou *et al.*, 2007], CMTL [Zhou *et al.*, 2011] and GMTL [Pu *et al.*, 2014]. All the methods are based on the same KDES features as ours. The regularization parameters are chosen by cross validation on the training set for all the MTL methods. In order to comprehensively assess the results, we also provide some FGC methods as baselines. Recently, several work exploits the techniques of segmentation and localization on object parts to obtain strong performance on the dataset, such as [Chai *et al.*, 2013; Gavves *et al.*, 2013]. These methods contain extra procedures, and the results highly depend on the quality of segmentation and localization. Thus, we leave them out of the comparison, however our advantage is complementary to their strength of modeling object parts.

The results of different methods on overall 200 classes are summarized in Table 1. Our result is obtained when $\lambda_c = 0.8$ and $\lambda_s = 0.1$. The performance is measured by accuracy. Adaptive Sharing achieves better result than baseline methods. In particular, we can observe that it clearly outperforms baseline MTL methods. Compared with the FGC methods, baseline MTL methods achieve marginal improvement, even worse performance. Complex intra-class variations and inter-class correlation incur negative transfer in these methods, which hurts the overall performance. In contrast, our method appropriately exploits shared information by jointly capturing the task relationships and individual disparities.

To further evaluate the performance among the MTL methods, we test them on a frequently used subset of CUB- 200-2010 dataset, Birds14. The subset is comprised of 14 species

Table 2: Architectures of the networks used for classification on CIFAR-100. The convolutional layer is denoted as "conv <receptive field>, <filters>". The max-pooling layer is denoted as "maxpool <region size>, <stride>".

input size	model A / A-AS	model B / B-AS
32	conv 5×5, 96	conv 3×3, 64
		conv 3×3, 64
	maxpool 3×3, 2	maxpool 2×2, 2
16	conv 5×5, 128	conv 3×3, 128
		conv 3×3, 128
	maxpool 3×3, 2	maxpool 2×2, 2
8	conv 5×5, 256	conv 3×3, 256
		conv 3×3, 256
	maxpool 3×3, 2	spp, {4, 2, 1}
-	FC, 2048	
-	FC, 2048	
-	FC, 100 / AS, 100	
-	softmax	

Table 3: Performance comparison of the four models on CIFAR-100. In the brackets are the improvements over the "no Adaptive Sharing" baselines.

Model	A	A-AS	B	B-AS
Test error (%)	37.20	$35.75_{(1.45)}$	33.17	$31.20_{(1.97)}$

birds, which are from two families: vireos and woodpeckers [Duan *et al.*, 2012]. Following [Duan *et al.*, 2012], the evaluation is measured by Average Precision (AP) on each class. We report the mean AP (mAP) as the overall performance of different methods in Table 1. Our result is obtained when $\lambda_c = 0.1$ and $\lambda_s = 0.01$. Adaptive Sharing achieves an outstanding result compared with these baseline methods. Moreover, we list the per-class results of our method in Fig. 2. As shown in the figure, the performance of our method is consistently better than CMTL and GMTL on almost all of the species. This further illuminates that our method benefits from appropriately sharing information, and hence can effectively improve the generalization performance.

4.2 Evaluation on Deep Neural Networks

The CIFAR-100 dataset [Krizhevsky, 2009] is composed of 32×32 color images belonging to 100 classes, with 50,000 images for training and 10,000 images for testing. We choose this dataset because there are a large number of classes but each one has a few samples, making it suitable for demonstrating the efficacy of sharing paradigm. The dataset is preprocessed by subtracting the mean image over the training set. We evaluate without any data augmentation.

Our Adaptive Sharing (AS) is independent of the deep architectures used. We investigate two different convolutional neural network architectures, as shown in Table 2. The model A is the one used in [Hinton *et al.*, 2012]. Inspired by [Simonyan and Zisserman, 2014], we develop a deeper network, model B, by replacing one 5×5 convolutional (conv) layer

Table 4: Test errors on CIFAR-100.

Method	Test error
Tree based Priors	
[Srivastava and Salakhutdinov, 2013]	36.85%
Network in Network [Lin *et al.*, 2014]	35.68%
Deeply Supervised [Lee *et al.*, 2014]	34.57%
dasNet [Stollenga *et al.*, 2014]	33.78%
Deeper Network (B)	33.17%
Adaptive Sharing (B-AS)	**31.20%**

Table 5: Test errors (%) of B-AS against parameter λ_c and ε.

ε \ λ_c	10^{-1}	10^{-2}	10^{-3}	10^{-4}
10^{0}	32.45	31.87	31.39	32.03
10^{-1}	32.61	31.81	31.48	31.84
10^{-2}	32.23	31.82	**31.20**	31.97

with a stack of two 3×3 conv layers. We also incorporate spatial pyramid pooling (spp) [He *et al.*, 2014] into the model B, where the pyramid configuration is 4×4, 2×2 and 1×1. All weights layers (except for the last Fully-Connected (FC) layer) are followed by the Rectified Linear Unit (ReLU). Dropout is applied to all the pooling layers and the first two FC layers, with the dropout ratios 0.25 and 0.5, respectively. We implement the Adaptive Sharing as a standalone layer, which can be integrated by replacing the last FC layer. We denote the models as A-AS and B-AS, respectively. Consequently, there are four models, A and B, as well as A-AS and B-AS for comparison.

Our implementation is based on the publicly available code of Caffe [Jia *et al.*, 2014]. We train the networks by applying stochastic gradient descent with a mini-batch size of 128 and a fixed momentum of 0.9. The training is regularized by weight decay (the ℓ_2 penalty factor is set to 0.004). Particularly, the parameter matrix $\mathbf{S}$ in Adaptive Sharing is regularized with ℓ_1 weight decay (the penalty factor is set to 0.0005). The learning rate is initialized to 0.001, is divided by 10 when the error plateaus.

The performance on the four models is shown in Table 3. With respect to model A, we achieve a coincide result as the one reported in [Hinton *et al.*, 2012]. By integrating our Adaptive Sharing, the model performance can be effectively enhanced (with 1.45% improvement). On the other hand, [Srivastava and Salakhutdinov, 2013] achieves marginal improvement (test error is 36.85%, with 0.35% improvement) by using the same network configuration. The method is also formulated in transfer learning framework. However, information sharing is strictly governed by the determination of group structure, which may hurt the performance due to hard partitioning. In contrast, our method does not strictly require the relatedness to satisfy certain structure. It is flexible to couple related classes, and the information can be appropriately transferred in our model. The clear improvement over [Srivastava and Salakhutdinov, 2013] (1.10%) demonstrates the

Table 6: Class groups based on the model B-AS.

Superclass	Classes
superclass 1	apple, bottle, mushroom, orange, pear, sweet pepper
superclass 2	bowl, can, clock, cup, lamp, plate, television
superclass 3	bed, chair, couch, keyboard, table, telephone, wardrobe
superclass 4	baby, boy, girl, man, woman
superclass 5	lawn mower, pickup truck, tank, tractor
superclass 6	bee, beetle, butterfly, caterpillar, cockroach, spider
superclass 7	fox, leopard, lion, skunk, squirrel, tiger, wolf
superclass 8	maple tree, oak tree, palm tree, pine tree, willow tree
superclass 9	dolphin, flatfish, ray, seal, shark, turtle, whale
superclass 10	bear, camel, cattle, elephant
superclass 11	cloud, forest, mountain, plain, road, sea
superclass 12	aquarium fish, trout
superclass 13	beaver, otter, porcupine, shrew, snail
superclass 14	bicycle, motorcycle
superclass 15	crab, crocodile, lizard, lobster, snake, worm
superclass 16	hamster, mouse, possum, rabbit, raccoon
superclass 17	orchid, poppy, rose, sunflower, tulip
superclass 18	bridge, bus, castle, house, streetcar, train
superclass 19	chimpanzee, dinosaur, kangaroo
superclass 20	skyscraper, rocket

Table 7: Class groups based on the model B.

Superclass	Classes
superclass 1	cockroach, crab, lion
superclass 2	bee, butterfly, orchid, poppy, rose, sunflower, sweet pepper, tulip
superclass 3	flatfish, ray, shark, shrew, turtle
superclass 4	kangaroo, leopard, raccoon, skunk, tiger, wolf
superclass 5	cattle, table
superclass 6	clock, house, lamp, skyscraper, telephone
superclass 7	apple, bear, beaver, camel, chimpanzee, elephant, otter, pear, seal
superclass 8	bus, lawn mower, motorcycle, pickup truck, tractor, train
superclass 9	bed, chair, couch, keyboard, snake, worm
superclass 10	maple tree, oak tree, palm tree, pine tree, porcupine, willow tree
superclass 11	dolphin, rocket, whale
superclass 12	bridge, forest, road, sea, streetcar, television
superclass 13	bottle, bowl, can, cup, orange, plate, wardrobe
superclass 14	mushroom, snail, squirrel
superclass 15	crocodile, dinosaur, lobster, trout
superclass 16	beetle, lizard, spider
superclass 17	baby, boy, girl, man, woman
superclass 18	aquarium fish, fox, hamster, mouse, possum, rabbit
superclass 19	bicycle, mountain, plain
superclass 20	castle, caterpillar, cloud, tank

advantage of Adaptive Sharing. It is worth noting that Adaptive Sharing can enhance the model A and B consistently, and the superiority is more significant in the deeper network B. This implies that such sharing paradigm is beneficial for discovering useful features.

In order to comprehensively confirm the effectiveness of our method, we compare with the previous state-of-the-art results, as shown in Table 4. By virtue of a deeper architecture, the model B (in Table 3) outperforms the published best result. However, the superiority is marginal. Our Adaptive Sharing (B-AS) further improves the result. A test error of 31.20% is achieved, which surpasses dasNet [Stollenga *et al.*, 2014] by 2.58%.

The balance between the shared part $\mathbf{C}$ and the specific part $\mathbf{S}$ can be regularized by the values of the parameters λ_s and λ_c. Due to many parameters in deep neural networks, we apply a simple strategy that specifies λ_s (and other parameters in the networks) with aforementioned empirical value, and study the effect of low rank parameters (i.e., the regularization factor λ_c and the approximation factor ε in (11)) on model performance. The test errors of B-AS with different values of λ_c and ε are summarized in Table 5. The results show that the model is insensitive to the factor ε, that similar performance can be obtained with a fixed λ_c. In contrast, the regularization factor λ_c plays a critical role in model performance. The best result is obtained when λ_c is set to 10^{-3}.

To investigate the power of our Adaptive Sharing in capturing the class relatedness, we make an analysis on the matrix $\mathbf{C}$ (corresponding to the shared features) of the model B-AS. We utilize $\mathbf{C}^T\mathbf{C}$ to represent the similarity matrix of all the classes in CIFAR-100, which is shown in Fig. 3. Darker color describes larger value in the similarity matrix (diagonal line denotes self-similarity). The matrix exhibits block-diagonal structure, indicating that related classes are encouraged to be coupled and share information. While hard partitioning is not applied in Adaptive Sharing, we adopt the Normalized Cut [Shi and Malik, 2000] to visualize the induced class groups of the similarity matrix, as shown in Table 6. For comparison, we also provide the class groups of the model B by adopting a similar operation on the weight matrix of the last FC layer,

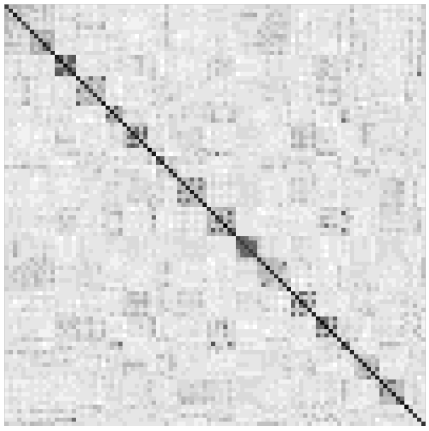

Figure 3: Similarity matrix ($\mathbf{C}^T\mathbf{C}$) of CIFAR-100.

and the result is shown in Table 7. It is obvious that the relatedness captured in Adaptive Sharing is more reasonable. In fact, the cluster relationship among classes is introduced in our method although there is no explicit partitioning as [Srivastava and Salakhutdinov, 2013] did.

We perform additional experiments on the ImageNet 2012 classification dataset [Russakovsky *et al.*, 2014], which is a challenging dataset with 1000 classes. We use the AlexNet architecture [Krizhevsky *et al.*, 2012] (based on Caffe training protocol [Jia *et al.*, 2014]) as baseline, which achieves 57.1% on top-1 accuracy and 80.2% on top-5 accuracy on the validation set, using the center crop. We integrate our Adaptive Sharing in the network by replacing the last FC layer. We apply the same setting of the parameters λ_c, λ_s and ε as in CIFAR-100, and set the other parameters the same as the baseline. Our model is trained on a single Tesla K40 GPU within two weeks. We obtain 57.7% on top-1 accuracy and 81.3% on top-5 accuracy on the validation set, where the improvements over the baseline are 0.6% and 1.1%, respectively. Due to the high baseline on top-5 accuracy, the improvement is more difficult than the one on top-1 accuracy.

Nevertheless, our model achieves more improvement on top-5 accuracy by virtue of appropriately transferring knowledge among classes.

5 Conclusion

In this paper, we present a novel adaptive sharing method for image classification. The shared information is selectively extracted and exploited to improve the generalization performance while simultaneously identifying the class-specific properties. We further integrate such adaptive sharing with deep neural networks. The outstanding performance on multiple challenging datasets verifies the effectiveness of such adaptive transfer. As a future direction, we are interested in leveraging such sharing paradigm to model the relation among the filters in deep neural networks.

Acknowledgments

This work was supported in part by National Basic Research Program of China (973 Program): 2012CB316400, in part by National Natural Science Foundation of China: 61332016, 61025011 and 61272326, in part by Grant of University of Macau (MYRG202(Y1-L4)-FST11-WEH). Z. Lin is supported by 973 Program of China (grant no. 2015CB352502), NSF China (grant nos. 61272341 and 61231002), and Microsoft Research Asia Collaborative Research Program. We would like to thank NVIDIA for GPU donation.

References

[Angelova and Zhu, 2013] A. Angelova and S. Zhu. Efficient object detection and segmentation for fine-grained recognition. In *CVPR*, 2013.

[Archambeau *et al.*, 2011] C. Archambeau, S. Guo, and O. Zoeter. Sparse Bayesian multi-task learning. In *NIPS*, 2011.

[Argyriou *et al.*, 2007] A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In *NIPS*, 2007.

[Argyriou *et al.*, 2008] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

[Beck and Teboulle, 2009] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[Bo *et al.*, 2010] L. Bo, X. Ren, and D. Fox. Kernel descriptors for visual recognition. In *NIPS*, 2010.

[Caruana, 1997] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[Chai *et al.*, 2012] Y. Chai, E. Rahtu, V. Lempitsky, L. Van Gool, and A. Zisserman. Tricos: A tri-level class-discriminative co-segmentation method for image classification. In *ECCV*, 2012.

[Chai *et al.*, 2013] Y. Chai, V. Lempitsky, and A. Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. In *ICCV*, 2013.

[Chen *et al.*, 2012] J. Chen, J. Liu, and J. Ye. Learning incoherent sparse and low-rank patterns from multiple tasks. *Trans. KDD*, 5(4):22, 2012.

[Deng *et al.*, 2014] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam. Large-scale object classification using label relation graphs. In *ECCV*, 2014.

[Duan *et al.*, 2012] K. Duan, D. Parikh, D. Crandall, and K. Grauman. Discorvering localized attributes for fine-grained recognition. In *CVPR*, 2012.

[Evgeniou and Pontil, 2004] T. Evgeniou and M. Pontil. Regularized multi-task learning. In *KDD*, 2004.

[Farrell *et al.*, 2011] R. Farrell, O. Oza, N. Zhang, V. Morariu, and T. Darrell. Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. In *ICCV*, 2011.

[Fei-Fei *et al.*, 2006] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *Trans. PAMI*, 28(4):594–611, 2006.

[Gavves *et al.*, 2013] E. Gavves, B. Fernando, C.G.M. Snoek, A.W.M. Smeulders, and T. Tuytelaars. Fine-grained categorization by alignments. In *ICCV*, 2013.

[Gong *et al.*, 2012] P. Gong, J. Ye, and C. Zhang. Robust multi-task feature learning. In *KDD*, 2012.

[He *et al.*, 2014] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014.

[Hinton *et al.*, 2012] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580*, 2012.

[Jalali *et al.*, 2010] A. Jalali, P. Ravikumar, S. Sanghavi, and C. Ruan. A dirty model for multi-task learning. In *NIPS*, 2010.

[Jia *et al.*, 2014] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv:1408.5093*, 2014.

[Kang *et al.*, 2011] Z. Kang, F. Grauman, and F. Sha. Learning with whom to share in multi-task feature learning. In *ICML*, 2011.

[Khan *et al.*, 2011] F. S. Khan, J. Van De Weijer, A.D. Bagdanov, and M. Vanrell. Portmanteau vocabularies for multi-cue image representation. In *NIPS*, 2011.

[Krizhevsky *et al.*, 2012] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012.

[Krizhevsky, 2009] A. Krizhevsky. Learning multiple layers of features from tiny images. *Technical report*, 2009.

[Lee *et al.*, 2014] C. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply supervised nets. In *NIPS Workshop*, 2014.

[Lin *et al.*, 2011] Z. Lin, R. Liu, and Z. Su. Alternating directionmethod with adaptive penalty for low rank representation. In *NIPS*, 2011.

[Lin *et al.*, 2014] M. Lin, Q. Chen, , and S. Yan. Network in network. In *ICLR*, 2014.

[Liu *et al.*, 2009] J. Liu, S. Ji, and J. Ye. Multi-task feature learning via efficient $\ell_{2,1}$-norm minimization. In *UAI*, 2009.

[Mei *et al.*, 2012] S. Mei, B. Cao, and J. Sun. Encoding low-rank and sparse structures simultaneously in multi-task learning. *Technical report*, 2012.

[Obozinski *et al.*, 2006] G. Obozinski, B. Taskar, and M.I. Jordoan. Multi-task feature selection. *Technical report*, 2006.

[Pu *et al.*, 2014] J. Pu, Y. Jiang, J. Wang, and X. Xue. Which looks like which: Exploring inter-class relationships in fine-grained visual categorization. In *ECCV*, 2014.

[Russakovsky *et al.*, 2014] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *arXiv:1409.0575*, 2014.

[Shi and Malik, 2000] J. Shi and J. Malik. Normalized cuts and image segmentation. *Trans. PAMI*, 22(8):888–905, 2000.

[Simonyan and Zisserman, 2014] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.

[Srivastava and Salakhutdinov, 2013] N. Srivastava and R. Salakhutdinov. Discriminative transfer learning with tree-based priors. In *NIPS*, 2013.

[Stollenga *et al.*, 2014] M. F. Stollenga, J. Masci, F. Gomez, and J. Schmidhuber. Deep networks with internal selective attention through feedback connections. In *NIPS*, 2014.

[Thrun and Pratt, 1998] S. Thrun and L. Pratt. *Learning to learn*. Kluwer Academic Publishers, Norwell, MA, USA, 1998.

[Welinder *et al.*, 2010] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. *Technical report CNS-TR-2010-001*, 2010.

[Yang *et al.*, 2009] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

[Yang *et al.*, 2012] S. Yang, L. Bo, J. Wang, and L. Shapiro. Unsupervised template learning for fine-grained object recognition. In *NIPS*, 2012.

[Yao *et al.*, 2012] B. Yao, G. Bradski, and L. Fei-Fei. A codebook-free and annotation-free approach for fine-grained image categorization. In *CVPR*, 2012.

[Yu *et al.*, 2005] K. Yu, V. Tresp, and A. Schwaighofer. Learning Gaussian processes from multiple tasks. In *ICML*, 2005.

[Zhou *et al.*, 2011] J. Zhou, J. Chen, and J. Ye. Clustered multi-task learning via alternating structure optimization. In *NIPS*, 2011.