Tranfer Learning in Large Scale Datasets

Huizhong Chen, Bowen Meng
Email: hchen2, bowenm@stanford.edu
CS229 Project Milestone

Ying Chang
Email: changy@stanford.edu
Co-work for CS 294a with Professor Daphne Koller

***Abstract*—Conventional image classification techniques aim to predict class labels by training a classifier with the provided training labels, but the internal relationship between classes has been ignored. In this project, we explore the feasibility of transferring knowledge between classes to help boost up the classification accuracy. Two transfer learning approaches have been studied, namely, instance transfer by jointly optimizing classifiers via grouping source and target training examples, and parameter transfer by exploring the relationship between classifier parameters. We demonstrate the parameter transfer scheme achieves a remarkably better performance compared to conventional image classification techniques and the relatively simple instance transfer approach.**

I. Introduction

To better reflect the richness of our visual world, several large-scale computer vision datasets have emerged recently, e.g., the 15-Class Scene dataset [1] [2], ImageNet [3] and SUN [4] dataset. Although thousands even millions of images are collected in these datasets, some classes only have a small number of instances due to the intrinsic long-tailed distribution of objects in the real world. For those classes with few instances, it is hard to obtain high-performance classifiers by treating and learning each individual class classifier independently because of the lack of training data. In fact, treating each class independently ignores a lot of infrastructures in the class space. For example, although ImageNet is organized according to WordNet [5] from the semantic aspect, the hierarchical structure sometimes correlates with the visual content, i.e., classes which are close to each other on the hierarchical tree are usually visually similar. So questions like what can be shared and transferred between classes and how to share/transfer so that the classification performance can be improved, give rise to the motivation of this work.

Our objective of this project is to improve the classification accuracy of the target class by migrating the knowledge from the source class, while not degrading the classifier performance for other classes in the dataset. The report is organized as follows. Section II describes our implementation of the 1-vs-all SVM for image classification without doing knowledge transfer, which serves as the baseline for comparison with our transfer learning performance. Then, in section III, we show that by jointly train an SVM using the source and target training examples, it is possible to increase the classification accuracy compared to the result by training target class alone, especially when the number of target training examples is scarce. However, the performance of instance transfer depends on certain choice of kernels, and more importantly, relies on the correlation between the source and the target classes. To overcome to drawbacks of instance transfer, in section IV, we propose a novel method by exploring similarity measures in SVM parameters to regularize the cost function of the target class SVM. Experimental results and discussions will be presented in section V. It is shown that our proposed parameter transfer method outperforms the no transfer baseline, as well as being more robust than the instance transfer scheme.

II. Content Based Scene Classification Using SVM

The 15-Class Scene dataset is adopted as the database to test image classification algorithms. Some sample images from the dataset are shown in Fig.1 In this section, we describe our baseline scene classification algorithm using Histogram of Oriented Gradients (HoG) features [6] and the implementation of the 1-vs-all SVM.

A. Dense Sampling HoG

Xiao et. al. [4] have reported that the densely sampled HoG features give the best scene classification performance on both the 15-Class Scene dataset and the SUN dataset. Therefore, HoG has been selected as our features, which will be feed to the SVM for image classification. The performance can be further boosted by aggregating other feature descriptors such as SIFT [7], GIST [8] and SSIM [9] to the HoG result, but in the interest of computational complexity we do not perform such analysis in this project.

The computation of HoG features follows the same approach as described in [4]. Histogram of oriented edge descriptors are densely extracted from the image at steps of 8 pixels. Then, every 2×2 neighboring HoG descriptors are concatenated to form a 124-dimensional descriptor. The descriptors are quantized into 300 visual words using k-means. Finally, three-level spatial histograms are computed on grids of 1×1, 2×2 and 4×4, which means for each image, the feature is a $300 \times (1^2 + 2^2 + 4^4) = 6300$ dimensional vector.

After extracting image features, kernels need to be computed and feed to the 1-vs-all SVM to perform scene classification. We have chosen two types of kernels, the linear kernel and the KL1 kernel (i.e. histogram intersection). The linear kernel is selected for computational convenience, while the choice of the KL1 kernel follows the definition of HoG similarity metric as described in [6].

Fig. 1. Sample images from the 15 Class Scene dataset

B. 1-vs-all SVM for classification

We employ LIBSVM [10] to implement a 1-vs-all SVM, with the aim to predict the class labels of testing examples. Note that there are more than two classes in the dataset, hence the 1-vs-all SVM has to loop through all the classes. The implementation is described as follows:

for $i = 1$ to (Number of classes) **do**
  Assign class i labels to $+1$, all remaining classes to -1;
  Train 1-vs-all SVM on training set;
  Compute confidence scores on testing set;
end for

After the SVM training and confidence score computation, for each testing data, we now have confidence scores whose size is equal to the total number of classes. The class label of a testing sample will then be predicted as the one which gives the highest confidence score among all classes.

III. Transfer Learning - Instance Transfer

In order to improve image classification performance, classes that have similar semantic meanings can be merged to train the SVM. This is called instance transfer because the instances of the source class is migrated to the target class. Rohrbach et. al. [11] have proposed using semantic relatedness between class labels to determine if knowledge transfer is advantageous, but in this project, the source and the target classes are manually assigned according to their labels. After identifying the source and the target, training examples from the source class are assigned to have the same label as the target class to jointly train the SVM. We implemented the SVM in a flexible and efficient way such that the training kernel matrix needs only be computed once for the whole dataset. During transfer learning, the corresponding kernels of the source class, the target class, and the remaining classes are selected from the kernel matrix of the whole dataset. In section V, the performance of scene classification with instance transfer is evaluated. We will see that instance transfer sometimes outperforms the baseline result when no transfer is carried out, but it is constrained by certain choices of SVM kernels, and the performance depends on the correlatedness between the source and the target. To overcome the limitations of instance transfer, we propose another approach by transferring knowledge among classifier parameters.

IV. Transfer Learning - Parameter Transfer

Besides instance transfer, it is also possible to transfer knowledge in the parameter domain by seeking the relationship between the source and the target classifier parameters. This type of transfer learning is called parameter transfer. Previously proposed framework and methods for multi-task learning are based on the assumption of the relatedness of the tasks. For example, Evgeniou et. al. [12] consider that the classifier parameters ω of all classes are close to some mean parameter ω_0. But the assumption does not include the information about how "close" each of the parameters are to the mean parameter ω_0, neither did it specify such an ω_0. In our work, we assume that the source class could help the target class in two ways: 1) when their classifier parameters are close/related, the source class parameters helps to pull the target class parameters close; 2) when their classifier parameters are far away/unrelated, then the source class parameters should push the target class parameters away towards better values. To sum up, the idea is that how much the source could assist the target is determined by the similarity of their classifier parameters.

The relationship in terms of ω_t and ω_s can be written as:

$$\omega_t = \omega_s + \nu$$

where ω_t and ω_s are the parameters for the target class and the source class respectively, and the sub-indices t and s denote the target class and the source class hereafter. w_s, functioning as the w_0 as described above, is an improvement Evgeniou's method [12] since it is based on the infrastructure of different classes rather than assumptions. The difference of the source and the target parameters is ν, which specifies the distances in different dimensions of the classifier parameter.

Therefore, the objective function of the SVM can be written as:

$$\begin{aligned} \min \quad J(\omega_t) &= \frac{1}{2}{\omega_t}^T\omega_t + \frac{1}{2}\nu^T diag(\lambda)\nu + C\sum_i \xi_i \\ &= \frac{1}{2}{\omega_t}^T\omega_t + \frac{1}{2}(\omega_t - \omega_s)^T diag(\lambda)(\omega_t - \omega_s) \\ &\quad + C\sum_i \xi_i \end{aligned} \tag{1}$$

$$\text{s.t.} \quad y_t^{(i)}(\omega_t^T x_t^{(i)} + b) \geq 1 - \xi_i \qquad \text{for } i = 1 \cdots m$$
$$\xi_i \geq 0 \qquad \text{for } i = 1 \cdots m$$

where $diag(\lambda)$ is a square matrix with its diagonal elements being $\lambda = [\lambda_1, \lambda_2, \cdots, \lambda_n]^T$. Here, $\lambda_i = \frac{\beta}{(\omega_{t,i}^{pre} - \omega_{s,i}^{pre})^2}$ and β denotes a weighting factor controlling how close we force the two group of parameters ω_t and ω_s to be. A larger β indicates a closer relation between the parameters ω_t and ω_s and vice versa. $\omega_{t,i}^{pre}$ and $\omega_{s,i}^{pre}$ are the pre-computed i-th parameters for the target class and source class by using ordinary SVM. Therefore, $(\omega_{t,i}^{pre} - \omega_{s,i}^{pre})^2$ is essentially the i-th empirical distance between the target and the source class, which will be employed as a prior knowledge for the following later transfer learning stage. As shown in 1,the empirical distance is used as the denominator so as to normalize the penalty term introduced by the assumption of parameter similarity. With the tool of Lagrangian, we can formulate the dual problem as:

$$\begin{aligned} \max \quad & \sum_i \alpha_i - \frac{1}{2}\alpha^T diag(Y_t) X_t^T diag(1+\lambda)^{-1} X_t diag(Y_t)\alpha \\ & - \alpha_s^T diag(Y_s) X_s^T diag(\frac{\lambda}{1+\lambda}) X_t diag(Y_t)\alpha \qquad (2) \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C \\ & \sum_i \alpha_i y_i^i = 0 \end{aligned}$$

where α is the dual optimizer for the same problem. X_t and X_s are the training features for the target class and the source class respectively.$Y_t = [y^{(1)}, \cdots, y^{(m)}]$ are the target class training labels and Y_s are the source class training labels. $diag(1+\lambda)^{-1}$ is the inverse of the square matrix whose diagonal is $1 + \lambda$. Mathematically, $diag(1+\lambda)^{-1} = diag(\frac{1}{1+\lambda})$. α_s^T is the pre-computed dual optimizer for the source class, and it is resulted by substituting the SVM equation $\omega_s = \sum_i \alpha_{s,i} y_s^{(i)} x_s^{(i)}$ into (1). Hence, we have the the dual optimization in the kernel form as in (2). Note that $X_t^T diag(1+\lambda)^{-1} X_t$ and $X_s^T diag(\frac{\lambda}{1+\lambda}) X_t$ can be treated as new kernels and can be computed efficiently. We call $X_t^T diag(1+\lambda)^{-1} X_t$ and $X_s^T diag(\frac{\lambda}{1+\lambda}) X_t$ the reweighted kernels. In our work, the computation of the reweighted kernels is carried out using linear kernel rather than "KL1" or other commonly used kernels, because $X_t^T diag(1+\lambda)^{-1} X_t$ and $X_s^T diag(\frac{\lambda}{1+\lambda}) X_t$ only have effective distance meanings in the linear form.

V. Experimental Results

In this section, the performance of both instance transfer and parameter transfer will be evaluated. We will show that although the relatively naive instance transfer scheme improves the classification performance under some circumstances, it suffers from two major drawbacks: 1) Not well generalizable to different types of kernels; 2) The choice of the source and target largely affects the classifier performance. On the other hand, the parameter transfer scheme has demonstrated its superior ability in overcoming these two drawbacks hence and we conclude parameter transfer is well suited for knowledge transfer between arbitrary classes.

A. Evaluation of Instance Transfer

Our experiment is performed on the 15 Class Scene dataset, where the classification accuracies of SVM with and without instance transfer have been evaluated. To study the effect of the size of the training set, the number of training examples for the target class varies from 1 to 100, whilst the number of training examples for the source and each of the remaining 13 classes is kept at 100. To eliminate the randomness of the experiment, each test is performed 10 times, every time using randomly sampled images to train and test the SVM.

1) Instance Transfer - Source and Target Closely Related: In our first experiment, the "MIT highway" class and the "MIT street" class are used as the source and the target respectively. These two classes are closely related so a better classification accuracy should be resulted from transfer learning. As depicted in Fig.2a, if the KL1 kernel is used, with the help of the source class, the scene recognition accuracy of the target class is significantly better when the target class has very few training samples. The target class recognition accuracy with and without the source converges as its number of training samples increases. This agrees with our intuition that transfer learning will be most beneficial when training data is scarce. However, for the linear kernel case, Fig.2a shows that instance transfer actually hurts the SVM performance. This implies that instance transfer may not generalize well to different types of kernels.

It is also worthwhile to learn the effect of instance transfer on the classification performance averaged over all the data classes. Ideally, transferring knowledge to the target class should not degrade the classification performance for other classes. Fig.2b plots the recognition accuracy averaged over all the training classes. For the KL1 kernel case, as illustrated in Fig.2b, instance transfer offers a better accuracy when the number of target training samples is small. However, as the target training set grows, SVM without instance transfer starts to give a higher accuracy. This is because as the number of target training examples increases, the knowledge from the source becomes less and less useful. In fact, the source can be regarded as a kind of noisy training examples for the target. At a certain point, when the negative effect of the source class outgrows the positive knowledge it offers, instance transfer can be harmful and degrades the average classification accuracy of all classes. For the linear kernel case, unfortunately, instance transfer always gives a worse performance compared to the no knowledge transfer baseline. The poor performance of instance transfer using the linear kernel again verifies our understanding that instance transfer does not generalize well to the linear kernel.

2) Instance Transfer - Source and Target Not Related: In section V-A1, we have observed that when the source and target classes are closely related, instance transfer sometimes offers a better performance than the baseline of not doing any transfer learning. It is interesting to further investigate

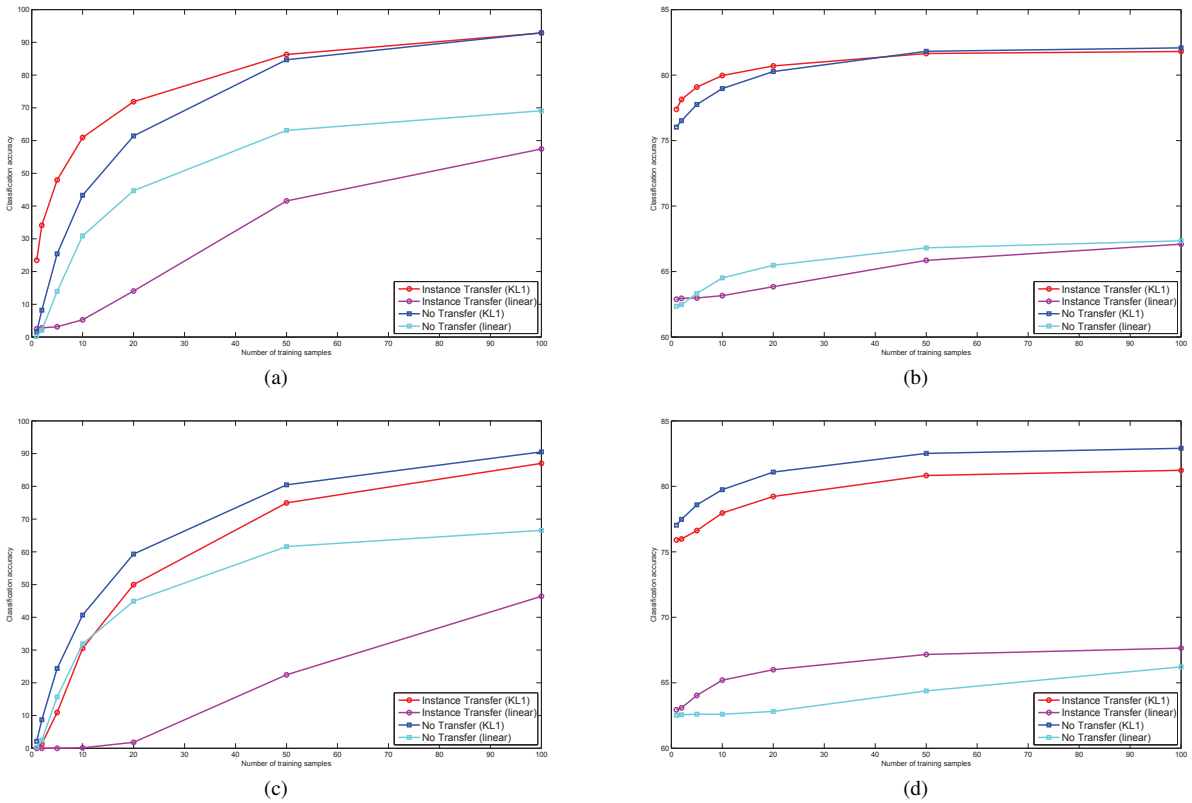

Fig. 2. Classification accuracy of instance transfer: (a) Target class accuracy, target=MIT street, source= MIT highway (b) All class accuracy, target=MIT street, source=MIT highway (c) Target class accuracy, target=MIT street, source=MIT coast (d) All class accuracy, target=MIT street, source=MIT coast.

the performance of transfer learning under the scenario when the source and the target are not closely related. Therefore, we select the source class to be "MIT coast" and the target class to be "MIT street", and perform the same experiment as described in section V-A1. As can be seen in Fig.2c and Fig.2d, for both the KL1 and the linear kernels, the instance transfer learning curve always underperforms the original no transfer learning curve. The degraded performance of instance transfer can be easily understood, as the source class examples now behave like "mislabelled" training samples. Since these "mislablled" training samples are grouped with the target class training samples to jointly train the SVM, the instance transfer performance is expected to be lower.

To summarize, under certain circumstances, instance transfer offers a better classification accuracy. However, instance transfer is very sensitive to the specific choice of the SVM kernel, as well as choice of the source and the target classes. In the next section, we will demonstrate that our parameter transfer scheme outperforms instance transfer in terms of these two aspects.

B. Evaluation of Parameter Transfer

Details of the parameter transfer algorithm have been described in section IV. As a comparison, the transfer learning scheme proposed in [12] has also been examined. The difference between our proposed method and Evgeniou's method is that, we compute the reweighted kernels by using the parameter empirical distance between the source and the target classifiers, whereas the work in [12] simply assigns a constant to the parameter distance. Fig.3 plots the learning curves for our proposed method, Evgeniou's method and the no transfer learning case. The experiment setup is the same as described in section V-A for the instance transfer scheme, but we only use the linear SVM kernel for the parameter transfer since the reweighted kernel has physical meanings only when the kernel is in the linear form. Fig.3a and Fig.3b simulate the scenario when source and the target are closely related, whilst Fig.3c and Fig.3d consider the case when the source and the target are unrelated. To study the effect of the weighting factor β in (1), learning curves are plotted for $\beta = 10, 100$ and 1000. As illustrated in Fig.3, regardless of the relationship between the source and the target classes, our proposed reweighted parameter transfer always outperforms the result when no transfer learning is conducted. Also note that given a certain choice of β, our method offers a better performance than the non-reweighted parameter transfer algorithm.

Recall that the instance transfer scheme is sensitive to: 1) The choice of kernel; 2) The choice of the source and target. Comparing our reweighted parameter transfer scheme to the relatively naive instance transfer scheme, it is obvious that reweighted parameter transfer overcomes these two major drawbacks. As can be seen in Fig.3, even though the linear SVM kernel is used, the classification accuracy for parameter

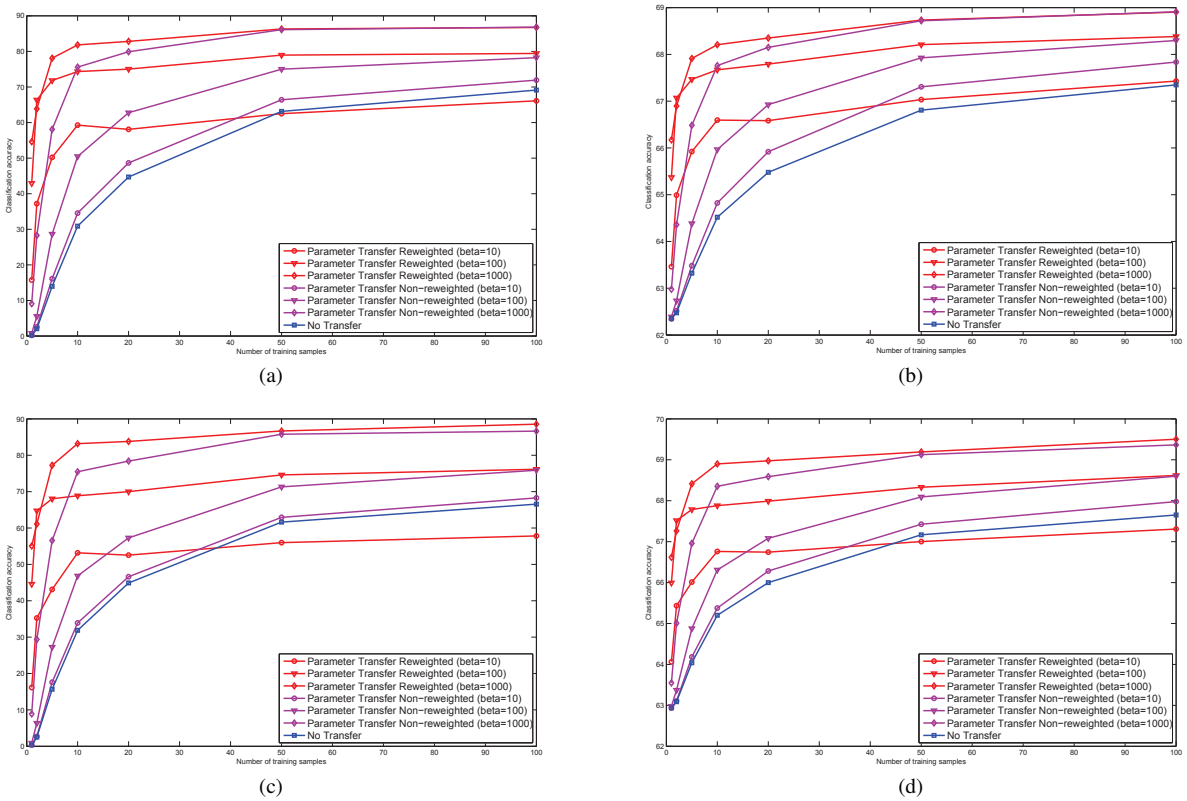

Fig. 3. Classification accuracy of parameter transfer: (a) Target class accuracy, target=MIT street, source= MIT highway (b) All class accuracy, target=MIT street, source=MIT highway (c) Target class accuracy, target=MIT street, source=MIT coast (d) All class accuracy, target=MIT street, source=MIT coast.

transfer still outperforms the no transfer learning baseline. More importantly, the reweighted parameter transfer is robust to arbitrary choice of the source and the target. In other words, given any combination of source and target classes, reweighted parameter transfer always offers a higher classification accuracy no matter whether the source and target are semantically closely related or unrelated.

VI. Conclusions

In this project, we demonstrated the usefulness of transferring knowledge in multi-task learning. Our proposed reweighted parameter transfer scheme provides a significantly improved performance over non-reweighted parameter transfer and no transfer. Also, reweighted parameter transfer is not sensitive of the choice of source and target classes and hence is more reliable than instance transfer.

At the current stage, our reweighted parameter transfer only works for the linear kernel since the parameter distance is physically more interpretable in the linear form. In future, it is interesting to further understand the parameter distance and investigate the feasibility to generalize our work to accommodate other types of SVM kernels.S

References

[1] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. II: 2169–2178.

[2] F. F. Li and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *CVPR*, 2005, pp. II: 524–531.

[3] J. Deng, W. Dong, R. Socher, K. L. L.-J. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[4] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *Conference on Computer Vision and Pattern Recognition*, 2010.

[5] Princeton university: Wordnet. [Online]. Available: http://wordnet.princeton.edu

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005, pp. 886–893.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[8] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, pp. 145–175, 2001.

[9] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, 2007, pp. 1–8.

[10] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[11] M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, and B. Schiele, "What helps where - and why? semantic relatedness for knowledge transfer," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2010, pp. 910 –917.

[12] T. Evgeniou and M. Pontil, "Regularized multi–task learning," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, pp. 109–117.