Multi-Task Multi-Sample Learning

Yusuf Aytar(✉) and Andrew Zisserman

Visual Geometry Group, Department of Engineering Science, University of Oxford,
Oxford, UK
yusuf@robots.ox.ac.uk

Abstract. In the exemplar SVM (E-SVM) approach of Malisiewicz *et al.*, ICCV 2011, an ensemble of SVMs is learnt, with each SVM trained independently using only a single positive sample and all negative samples for the class. In this paper we develop a *multi-sample learning* (MSL) model which enables joint regularization of the E-SVMs without any additional cost over the original ensemble learning. The advantage of the MSL model is that the degree of sharing between positive samples can be controlled, such that the classification performance of either an ensemble of E-SVMs (sample independence) or a standard SVM (all positive samples used) is reproduced. However, between these two limits the model can exceed the performance of either. This MSL framework is inspired by multi-task learning approaches.

We also introduce a multi-task extension to MSL and develop a *multi-task multi-sample learning* (MTMSL) model that encourages both sharing between classes and sharing between sample specific classifiers within each class. Both MSL and MTMSL have convex objective functions.

The MSL and MTMSL models are evaluated on standard benchmarks including the MNIST, 'Animals with attributes' and the PASCAL VOC 2007 datasets. They achieve a significant performance improvement over both a standard SVM and an ensemble of E-SVMs.

Keywords: Multi-task learning · Exemplar SVMs

1 Introduction

A number of recent papers in computer vision [11,15] have explored the use of a mixture of linear SVM classifiers [13,30] and locally linear SVMs [12,19]. In cases where there is a large diversity in positive training samples, for example articulations of a human [15] or viewpoints of an object [11], superior performance is achieved by multiple linear classifiers, compared to limiting the classifier to a single linear SVM. This is because each linear SVM can learn a template for a tight cluster ('components' or 'aspects') of visual appearances. Motivated by this success, Malisiewicz *et al.* [23] investigated the limit of the idea and introduced the exemplar SVMs (E-SVMs), where a SVM is trained for each positive sample together with all the negative samples, and the final classifier is defined as an ensemble of exemplar SVMs.

L. Agapito et al. (Eds.): ECCV 2014 Workshops, Part III, LNCS 8927, pp. 78–91, 2015.
DOI: 10.1007/978-3-319-16199-0_6

In this paper we introduce models to explore the spectrum between a single linear SVM and an ensemble of E-SVMs. The single linear SVM may not have the capacity to model all per sample variations, but has the possibility of generalizing across multiple positive samples. At the other end of the spectrum, an ensemble of E-SVMs can certainly accommodate sample specific variation, but has no possibility of learning across positive samples, since each E-SVM is a sample specific classifier learnt independently from a single positive sample. We introduce here *multi-sample learning* (MSL), for jointly learning multiple E-SVMs. This has the flexibility to travel between the two ends of the learning spectrum (i.e. single SVM and ensemble of E-SVMs) without any extra cost over E-SVMs. The advantages of MSL are that (i) a sweet spot can be chosen between the two ends which improves classification performance, and (ii) compared to a mixture of linear SVMs, the formulation is convex.

The MSL formulation is inspired by multi-task learning (MTL) [5,10,26] approaches. In MTL, different classification tasks are jointly regularized, though in general only a loose relation between tasks is encouraged as the tasks might be very different from each other. In contrast, in MSL, sample specific classifiers for the same class are learnt simultaneously and the classifiers can be very close to one another. Thus, depending on the amount of similarity (or diversity) between the positive samples, the coupling between the classifiers can be encouraged more strongly. While diversity in the positive training samples are modeled with the sample specific classifiers, common features are favoured by joint regularization of these classifiers.

Moreover, we also introduce a multi-task extension to MSL which is termed *multi-task multi-sample learning* (MTMSL). This model encourages the sharing between classes and between sample specific classifiers within each class.

First, we present the formulations for MSL and MTMSL. Then, we describe the optimization procedure and implementation details. Finally, we illustrate the power of MSL and MTMSL on three example datasets, and discuss possible extensions to these methods.

2 Multi-Sample Learning

In this section we define the MSL objective for learning sample specific classifiers in a joint regularization framework. Assume we have a binary classification problem where N is the number of training samples, $X = \{x_i\}_{i=1}^N$ are the features, and $Y = \{y_i\}_{i=1}^N$ are the corresponding binary labels chosen from the set $y_i \in \{-1, 1\}$. Then the standard SVM objective for solving this classification problem is:

$$\min_{w} \quad \lambda ||w||^2 + \sum_{i}^{N} \max\left(0, 1 - y_i(w^\intercal x_i)\right) \tag{1}$$

where w is the classifier vector, λ controls the trade off between the hinge loss and regularization, and the bias term is included by appending a constant number to the end of each x_i and extending the w vector accordingly.

In contrast to the 'classical' SVM, where all positive samples are used to train a single vector w, MSL increases the capacity of the model by defining sample specific classifiers for each positive sample. However, unlike exemplar SVMs [23], in order to enable sharing between classifiers, MSL jointly regularizes the sample specific classifiers such that the objective can be tuned to behave as an SVM or ensemble of exemplar SVMs, or any point in between. The formulation is:

$$\min_{u,\mathbf{w}} \quad \lambda ||u||^2 + \beta \sum_{i \mid y_i=1}^{N} ||w_i - u||^2 + \sum_{i \mid y_i=1}^{N} L_e(w_i; X, Y) \tag{2}$$

$$\text{where} \qquad L_e(w_i; X, Y) = \max\left(0, 1 - w_i^{\mathsf{T}} x_i\right) +$$

$$\tfrac{1}{N^+} \sum_{j \mid y_j \neq 1}^{N} \max\left(0, 1 + w_i^{\mathsf{T}} x_j\right), \tag{3}$$

N^+ is the number of positive samples, u is the shared base vector, and the w_i's are sample specific classifiers defined for each positive sample. L_e represents the hinge loss for the given exemplar SVM. The hyperparameters λ and β control the trade off between the hinge loss and regularization, as well as the balance between individual sample specific regularization and joint regularization. With the appropriate setting of the hyperparameters (2) will converge to a classical SVM or an ensemble of exemplar SVMs.

As $\beta \to \infty$, the regularizer $\sum_i^N ||w_i - u||^2$ acts as the hard constraint $w_i = u, \forall i$. Thus each w_i is forced to be same as u and equation (2) becomes the classical SVM formulation (1). Note that in the loss function (3) the loss coming from negative samples are multiplied with $\frac{1}{N^+}$, this ensures exact equivalence with the SVM formulation in (1) since L_e is summed over each positive sample exactly N^+ times.

As $\lambda \to \infty$, u will be forced to a zero vector, the regularization term $\sum_i^N ||w_i - u||^2$ will become $\sum_i^N ||w_i||^2$, and the formulation is equivalent to learning each sample specific classifier individually (i.e. an ensemble of exemplar SVMs). In this case the multiplier $\frac{1}{N^+}$ for the negative loss terms becomes a balancing factor between a single positive and many negative samples. Stronger weighting for the positive loss term is also applied in the ensemble of exemplar SVMs [23] setting and is noted as a factor for improving the success of E-SVMs.

Note that the formulation (2) is convex and the global optimum can be found through standard convex optimization methods (section 4). [18] also uses a similar convex formulation, though targeting dataset bias.

Discussion. There are two main types of formulations for multi-task learning. In the first group, classifier vectors for each task are coupled by minimizing the Frobenius norms of the classifier vector differences [10,22,25] or by sharing a common prior [7,21,27]. In the second, the model parameters are generated from a common latent feature representation which is provided by different forms of nuclear norm regularization [1–5,24].

MSL encourages joint learning over samples through a shared vector u, in a similar manner to the multi-task learning framework of [10] from the first group. Different forms and anaysis of this particular type of regularization are investigated in [9,10] thoroughly including analysis of the dual form and the kernelization.

Considering that $w_{mean} = \frac{1}{N^+}\sum_i w_i = \arg\min_u \sum_i ||w_i - u||^2$, the regularizer that encourages the sharing in MSL, i.e. the term $\sum_i ||w_i - u||^2$, can also be written as $\sum_i ||w_i - w_{mean}||^2$ if there is no penalization on the norm of u (i.e. $\lambda = 0$). Since:

$$\sum_i ||w_i - w_{mean}||^2 = \frac{1}{2N^+}\sum_{i,j} ||w_i - w_j||^2 \tag{4}$$

it can be seen that this corresponds to a fully connected pairwise regularization structure between the sample specific classifiers, Therefore we can also write the regularization term as $\sum_{i,j} ||w_i - w_j||^2$, however in this form we lose the flexibility of imposing additional penalization on the shared vector (since there is no longer a term in u).

Convex approaches [1,3,4,24] from the second group, which are based on nuclear norm regularization, can also be used for encouraging the task relatedness. However, this is not suitable for our problem as we now discuss. The nuclear norm regularization induces low rank solutions and encourages the classifiers to be composed from a smaller set of latent basis vectors. It can be applied to softly enforce joint learning between the sample specific classifiers using a formulation such as:

$$\min_W \quad \lambda||W||_* + \sum_{i\,|\,y_i=1}^{N} L_e(w_i; X, Y) \tag{5}$$

where the columns of the matrix W are sample specific classifiers w_i, and λ controls the trade off between regularization and hinge loss. Since the nuclear norm encourages low rank solutions for the matrix W, with a sufficiently strong regularization (i.e. very large λ) a rank 1 solution for W can be obtained. If W is rank 1, then each w_i will have the same direction in the feature space (their magnitudes may differ but this does not effect the *ranking* order of test samples). Thus by a heavy nuclear norm penalization of W, each w_i will converge to a single classifier vector, in a similar manner to the convergence of MSL (2) to the classical SVM limit as $\beta \to \infty$. However, since $||W||_F \leq ||W||_*$, heavily penalizing $||W||_*$ also imposes a strong l_2 regularization on each w_i. The outcome is that w_i becomes over regularized (i.e. a very small magnitude vector), and consequently the performance drastically decreases. Therefore, using nuclear norm regularization, it is not possible to converge to a single classifier solution without a substantial loss in performance. This is the reason that we based the MSL on the first type of multi-task learning, rather than the second.

3 Multi-Task Multi-Sample Learning

In this section multi-task learning is incorporated with MSL in a joint formulation. This method again builds on the regularized multi-task learning approach of [10] which encourages sharing between tasks by minimizing the squared l_2 norms of classifier vector differences.

Unlike the binary classification problem, here we have multiple classes and the objective is to solve either a multi-class classification problem or multiple one-versus-all classification tasks trained simultaneously. In the first place we introduce a multi-class classification formulation and then describe learning multiple one-versus-all classifiers jointly.

In the **multi-class classification** setting, each sample belongs to a single class and the goal is to classify test samples into one of the existing classes. In the MSL formulation we have a single u which is shared across all sample specific classifiers w_i. In multi-task multi-sample learning (MTMSL), we have multiple u's, one for each class denoted as u_t. In addition to regularizing sample specific classifiers with the shared base vector u_t as in (2), we additionally regularize all the u_t's with another shared vector v which encourages sharing between u_t's. The formulation for MTMSL for the multi-class classification problem is:

$$\min_{v,\mathbf{u},\mathbf{w}} \quad \gamma||v||^2 + \lambda \sum_{t}^{T} ||u_t - v||^2 + \qquad (6)$$
$$\beta \sum_{i}^{N} ||w_i - u_{c(i)}||^2 + \sum_{t}^{T} \sum_{i \mid y_i^t = 1}^{N} L_e(w_i; X, Y^t)$$

where the w_i's are the sample specific classifiers, T is the number of classes, $c(i)$ is the class index of the i^{th} training sample, y_j^t is the binary label of the j^{th} sample for the class t, and similarly $Y^t = \{y_j^t\}_{j=1}^N$ is the set of binary labels for class t. The hyperparameters γ, λ and β determine the behavior of the formulation:

- As $\gamma \rightarrow \infty$, v will be forced to a zero vector, consequently the regularization term $\sum_t^T ||u_t - v||^2$ becomes $\sum_t^T ||u_t||^2$ and the formulation (6) is equivalent to T separate MSL (2) formulations each learning a classifier independently (i.e. there is no sharing across classes).
- As $\beta \rightarrow \infty$, each w_i will be forced to be equal to its class level shared vector $u_{c(i)}$, thus the formulation converges to the multi-task learning objective introduced in [10] (i.e. there is no multi-sample sharing).
- As both $\gamma \rightarrow \infty$ and $\beta \rightarrow \infty$, formulation learns T individual SVMs (1), one for each particular class (i.e. no multi-task or multi-sample sharing).

In the **multiple one-versus-all classification** setting, each training sample can have none (i.e. background) or several class labels, and the target is to classify the test sample as positive or negative for each class separately. With a slight change in the formulation MTMSL can support this setting:

$$\min_{v,\mathbf{u},\mathbf{w}} \quad \gamma||v||^2 + \lambda \sum_t^T ||u_t - v||^2 + \tag{7}$$
$$\beta \sum_t^T \sum_{i \mid y_i^t = 1}^N ||w_i^t - u_t||^2 + \sum_t^T \sum_{i \mid y_i^t=1}^N L_e(w_i^t; X, Y^t)$$

where w_i^t is the sample specific classifier of the i^{th} sample for class t. Note that both formulations (6) and (7) are convex.

Discussion. In the multi-task setting, unlike MSL where we might need strong coupling between E-SVMs, a nuclear norm based regularization [1,3,4,24] can also be used for encouraging the class-level task relatedness since we don't need very strong coupling between class level classifiers.

4 Optimization and Implementation Details

In this section we describe the optimization procedure used for minimizing our objective functions and the calibration of E-SVMs.

Since both MSL (2) and MTMSL (6, 7) are convex problems they can be minimized globally using convex optimization techniques. Particularly we use stochastic subgradient descent algorithm for optimizing our objectives. The optimization procedure will be described on the formulation (6) and it can be easily adapted to the other described formulations.

For convenience we cast the objective in (6) as:

$$\min_{v,\mathbf{\Delta u},\mathbf{\Delta w}} \quad \gamma||v||^2 + \lambda \sum_t^T ||\Delta u_t||^2 + \beta \sum_i^N ||\Delta w_i||^2 +$$
$$\sum_t^T \sum_{i \mid y_i^t=1}^N L_e(v + \Delta u_{c(i)} + \Delta w_i; X, Y^t) \tag{8}$$

At each iteration an E-SVM w_i and a sample x_j are randomly selected and the parameters $v, \Delta u_t, \Delta w_i$ are updated using the subgradients below:

$$v' = \gamma v - L_{ij} x_j, \qquad \Delta u_t' = \lambda \Delta u_t - L_{ij} x_j,$$
$$\Delta w_i' = \beta \Delta w_i - L_{ij} x_j$$

$$L_{ij} = \begin{cases} -1, \mathit{if} \quad y_j = -1 \quad \mathit{and} \quad w_i^\top x_j > -1 \\ 1, \quad \mathit{if} \quad i = j \quad \mathit{and} \quad w_i^\top x_j < 1 \\ 0, \quad \mathit{otherwise} \end{cases}$$

A decreasing learning rate is used inverse proportional to the iteration number.

One important step for the success of ensemble of E-SVMs [23] is the post calibration of sample specific classifiers. Even though learning E-SVMs jointly in

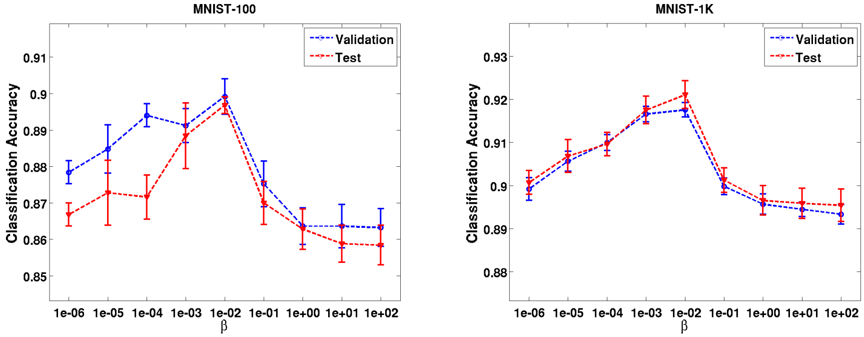

Fig. 1. The effect of the hyperparameter β in MSL on the MNIST dataset. The hyperparameter λ is fixed and the performance on both validation and test sets are shown. The multi-class classification accuracy as a function of the hyperparameter β is displayed. With a large enough β, the MSL gives the same result as the single class SVM. Moving towards the single class SVM (from left to right) the performance increases and then decreases back. Thus for an optimum β MSL outperforms both ensemble of E-SVMs and single class SVM.

the MSL framework provides a certain level of calibration, we also apply a post calibration step on a validation set such that the responses of each w_i on the validation set has zero mean and unit variance. The final classification score for any sample x is then obtained by a max over the calibrated E-SVMs weighted by the individual confidences:

$$f(x) = max_i^{N^+} c_i \frac{w_i^\mathsf{T} x - \mu_i}{\sigma_i}, \tag{9}$$

where μ_i is the mean and the σ_i is the standard deviation of the scores of w_i on the validation set, and c_i is the confidence of the w_i measured as the average precision (AP) of w_i evaluated on the validation set.

5 Experiments

In this section we present evaluation of our methods on three datasets: (a) MNIST digits dataset, (b) Animals with Attributes dataset [20], (c) PASCAL VOC 2007 dataset [8]. Datasets are separated into training, validation and test sets. The methods are trained on the training set and the hyperparameters λ, β, γ and calibration are determined using the validation set. The following methods are compared: SVM, ensemble of exemplar SVMs (EE-SVM), the proposed multi-sample learning (MSL), multi-task learning of [10] (MTL), and the proposed multi-task multi-sample learning (MTMSL)

In the first two experiments (i.e. MNIST and Animals with Attributes), the methods are evaluated on a *multi-class classification* setting, where the task is to classify each test sample into one of the existing classes. All the experiments are conducted 5 times with 5 different random selections of training, validation and test sets from the entire dataset. The mean multi-class classification accuracy and one standard error (i.e. defined as standard deviation of accuracies scaled by $1/\sqrt{5}$) is reported for each method. The PASCAL VOC 2007 dataset is particularly used for category detection experiments, where the task is to identify if each test sample belongs to the target category or not.

Since SVM, EE-SVM and MSL are binary classifiers, in order to use them in a multiclass classification setting, a one against all classifier is trained for each class using each method. The SVM classifiers are calibrated such that each one will give zero mean unit variance score on the validation set. For EE-SVM and MSL, each sample specific classifier is calibrated as described in (9) and class level scores are obtained. The final classification is performed by classifying each test sample into the maximum scoring class.

5.1 MNIST Dataset

MNIST dataset consists of 70K samples of 10 handwritten digits and the task is to classify each test sample into one of the digit classes. We extracted two subsets from this dataset. The first subset is extracted using the exact same setting described in [16]. The images are preprocessed with PCA and the dimensionality is reduced to 64 so as to retain $\sim 95\%$ of the total variance. This subset, referred as *MNIST-100*, consists of 100 training, 50 validation and 50 test samples per class. In order to observe the results for larger number of training samples we also evaluated on a second subset, referred as *MNIST-1K*, which uses 1000 training, 500 validation and 500 test samples per class.

Table 1 displays the classification accuracy results comparing all the methods on MNIST dataset. Using the same setting with [16] on MNIST-100, our SVM baseline is slightly better than their reported results. Since the randomizations might be different slight variations are expected. Among the binary classification methods MSL performs significantly better than the SVM and EE-SVM. It achieves at least 2% improvement over the other two methods on both subsets. In order to better visualize the behavior of MSL, we display the performance of the method as a function of the β parameter. Note that as $\beta \to \infty$, MSL converges to the SVM solution, and for very small β values it behaves closer to EE-SVM (it would behave exactly as EE-SVM if $\lambda = \infty$). As is shown in figure 1 on both of the subsets, moving from EE-SVM to SVM (from left to right) the performance increases and then decreases, finally reaching the SVM solution (large β values). This result demonstrates that combining the generalization of SVM and specification of EE-SVM can lead to much better results. With only 100 positive training samples per class, MSL reaches the accuracy of an SVM trained with 1000 positive samples (see table 1).

Note that the EE-SVM performance is not as good as SVM. In [23] it is clearly stated that in order to obtain a good performance from EE-SVM method each E-SVM needs to be trained against a very large number of negative training samples. Unfortunately in the classification problems we don't have as many negative samples as we have in object detection tasks where the negatives are unlimited (i.e. any subwindow of any image). This observation from [23] explains

Table 1. The multi-class classification accuracy comparison of methods on MNIST dataset. Note that MSL with 100 positive samples per class (MNIST-100) performs as well as SVM with 1000 positive samples per class (MNIST-1K). Note, the first row shows the individual task learning results from [16], and the MTL result in the first row is the learned grouping MTL of [16].

	SVM	**EESVM**	**MSL**	**MTL**	**MTMSL**
Animal-19	33.25±0.29	13.54±0.48	34.60±0.25	34.64±0.28	34.82±0.27
Animal-43	15.52±0.35	5.79±0.23	16.56±0.31	16.38±0.18	16.71±0.36

Table 2. The multi-class classification accuracy comparison of methods on Animal dataset

	SVM	**EESVM**	**MSL**	**MTL**	**MTMSL**
MNIST-100 of [16]	84.10±0.30*	n/a	n/a	84.80±0.30*	n/a
MNIST-100	85.84±0.55	78.12±0.99	89.68±0.22	85.72±0.47	89.44±0.16
MNIST-1K	89.57±0.37	82.70±0.40	92.10±0.32	89.55±0.34	92.04±0.32

the inferior behavior of EE-SVM method in our experiments. Nevertheless when learnt jointly, as in MSL, ensemble of these sample specific SVMs manages to outperform SVM solution.

As [16] noted as well, multi-task learning for MNIST dataset doesn't help much for improving the classification accuracy. Nevertheless, MTMSL method clearly outperforms the MTL approach for both of the subsets of MNIST dataset.

5.2 Animal Dataset

Animals with Attributes dataset [20], which will be referred as Animal dataset from here on, consists of 50 animal classes and $\sim 30K$ samples in total. For each sample, 2000 dimensional SIFT bag of words (BOW) features are kindly provided by the dataset creators [20]. As a preprocessing step we reduced the dimensionality of the features from 2000 to 500 using PCA. Some classes in the dataset have a small number of samples. In order to analyse the performance with different number of classes and samples we extracted two subsets from the Animal dataset. *Animal-43* subset consists of 43 classes which have more than 300 samples, and *Animal-19* consists of 19 classes which have more than 700 samples. *Animal-43* organized as 100 training, 100 validation and 100 test samples per class, and *Animal-19* organized as 500 training, 100 validation and 100 test samples per class. Similar to the previous problem the task is a multi-class classification problem and the same settings are used for calibration and evaluation.

The results on the Animal dataset is shown in table 2. Similar to the MNIST experiments MSL approach significantly outperform the EE-SVM and SVM by a margin of at least 1% improvement. Figure 2 shows the performance of MSL as a function of β parameter. It shows a similar behavior to MNIST experiments and gives an optimum result somewhere in between EE-SVM and SVM. Note

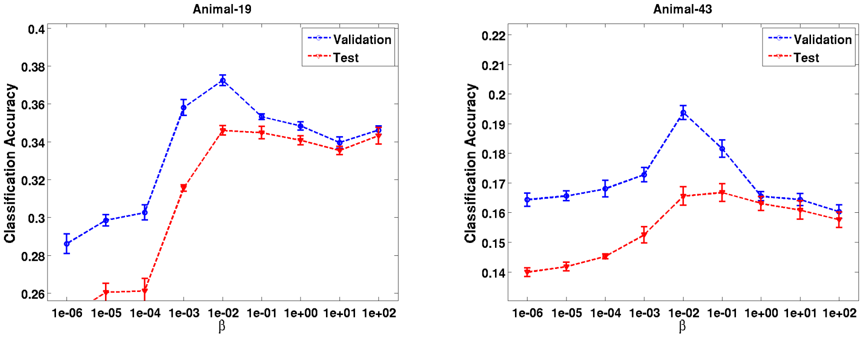

Fig. 2. The effect of the hyperparameter β in MSL on the Animal dataset. Note the increase in performance before reaching to the single class SVM limit (i.e. $\beta = 1e + 2$). See caption of figure 1.

the performance gap between the validation and test sets in figure 2. This gap suggests that we need more training samples for more stable results. Nevertheless it doesn't change the behavior of β parameter. Since calibration is performed on the validation set, MSL gives a better performance on the validation set compared to the test set. EE-SVM has a similar performance behavior as in MNIST experiment due to the reason explained in the previous section.

MTMSL only had a small gain over MSL for this dataset. As shown in table 1, MTL improves the results $\sim 1\%$ over the SVM result on both of the subsets. And MSL, which doesn't even use the multi-task learning or task relations, performs as well as MTL in the Animal-19 subset and better than MTL in the Animal-43 subset. Similar to the MNIST experiments, the multi-task extension of MSL, i.e. MTMSL, outperforms MTL on both of the subsets.

5.3 PASCAL VOC Category Detection

These experiments are performed on PASCAL VOC2007 dataset which contains 9,963 images. The dataset is arranged as 2501 training, 2510 validation and 4952 testing images. We picked bicycle, motorbike, horse and cow classes for our detection experiments as their side-facing examples have similar aspect ratios. For each category the mean bounding box(BB) is computed by taking the mean of each coordinate seperately across all the positive BBs belonging to the category. Then all the positive BBs are warped to the mean BB as we need the training samples to have same feature dimensionality. Histogram of oriented gradients (HOGs) [6] are used as the features. Training and validation is performed using all positive side-facing examples of the category together with 5000 random negative BBs cropped from the negative images. Similarly tests are performed on all positive side-facing examples and 20K negative BBs from the test set of PASCAL VOC2007 dataset. In these experiments MSL is compared against SVM and EE-SVM. As is shown in table 3, MSL constantly outperformed EE-SVM and SVM results.

Table 3. Average Precision results on side-facing category detection experiments. Evaluations are performed on all positive (side-facing) instances of the particular class and 20K negative instances extracted from PASCAL VOC 2007 test set.

	SVM	**EESVM**	**MSL**
bicycle	84.25	61.88	85.09
motorbike	75.81	18.56	76.36
horse	82.97	15.29	83.77
cow	70.14	14.53	70.84

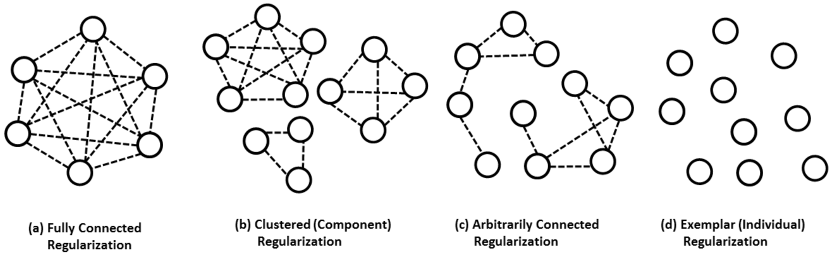
(a) Fully Connected Regularization
(b) Clustered (Component) Regularization
(c) Arbitrarily Connected Regularization
(d) Exemplar (Individual) Regularization

Fig. 3. Different models of joint regularization can be explored via regularization graphs. Each node represents a sample specific classifier and the links represent the weights of the joint regularization terms $||w_i - w_j||^2 \quad \forall i, j$ This paper particularly explores a type of fully connected regularization displayed in (a) with different levels of uniform weights on the edges which can be thought as springs. As the weight of edges increase classifiers are all forced to be as close as possible which in the limit reaches to a single class SVM, or if the weights become looser the classifiers become independent and in the limit reaches to the ensemble of exemplar svms displayed in (d). However, in between these two ends there are many other structural choices of the regularization graph to be explored as displayed in (b) and (c).

6 Conclusion and Future Directions

In this paper we introduced the multi-sample learning framework which combines the generalization ability of SVM with specialization property of EE-SVMs and provides a balanced learning framework which can travel between the two ends of the learning spectrum (i.e. SVM and EE-SVM).

We extended our approach to multi-task multi-sample learning which enables sharing between the classes as well as the sample specific classifiers within each class. By setting the hyperparameters appropriately, MTMSL can be tuned to behave as multiple SVMs, multiple EE-SVMs, multiple SVMs with MTL, multiple EE-SVMs with MTL or any sweet spot between these endpoints. We presented significant performance improvements in two datasets with varying sample sizes for both the MSL and MTMSL approaches.

Some recent MTL approaches take account of the task relationships in the MTL formulation [9,14,16,17,28,29] using the structure between tasks in the regularization. Relationships of this type can also be applied to MSL, We can define a joint regularization graph via defining relations between classifier pairs, and regularize the sample specific classifiers accordingly. We sketch this extension here.

Assuming that there is no penalization on the norm of u, then, as already mentioned in section 2, $\sum_i ||w_i - u||^2$ can be re-cast as pairwise difference regularizations $\frac{1}{2N^+} \sum_{i,j} ||w_i - w_j||^2$. If we represent the joint regularization relations as a graph whose nodes are the sample specific classifiers, then the regularization term $\sum_{i,j} ||w_i - w_j||^2$ corresponds to a fully connected graph structure (see figure 3). Furthermore, if we introduce weights for the joint regularization terms as $\sum_{i,j} A_{ij} ||w_i - w_j||^2$ where A encodes the graph structure, then the fully connected regularization becomes a special case of this new regularization term where $A_{ij} = 1, \forall i, j$. Subsequently we can encode any graph structure (e.g. clusters, hierarchies, or arbitrary regularization relations) by setting the adjacency matrix A accordingly. A few example structural choices of A are displayed in figure 3. Assuming that the relation matrix A is non-negative, then the regularizer will be convex. A regularization term can also be represented in the spectral form as below:

$$\sum_{ij} A_{ij} ||w_i - w_j||^2 = trace(WLW^\intercal) \tag{10}$$

$$where \quad L = D - A, \quad D_{ii} = \sum_{j \mid i \neq j}^{N^+} A_{ij}$$

where L is the graph laplacian of the regularization graph and the columns of the matrix W are sample specific classifiers w_i. The regularizer (10) is biconvex in L and W and it can be optimized by fixing one and optimizing the other iteratively.

Learning both classifiers and the graph structure from the data by additionally imposing regularizers on L opens many other possibilities of joint regularization. For instance, if we perform nuclear norm regularization on L it will provide us sparsity in the eigenvalues (same with the singular values in this case) of L. Since it is known that the number of zeros in the eigenvalues of the graph laplacian L defines the number of connected components in the graph, we naturally obtain a convex regularizer that encourages automated clustering of sample specific classifiers that will be jointly regularized.

References

1. Amit, Y., Fink, M., Srebro, N., Ullman, S.: Uncovering shared structures in multiclass classification. In: ICML, pp. 17–24 (2007)
2. Ando, R.K., Zhang, T.: A framework for learning predictive structures from multiple tasks and unlabeled data. Journal of Machine Learning Research **6**, 1817–1853 (2005)
3. Argyriou, A., Evgeniou, T., Pontil, M.: Multi-task feature learning. In: NIPS, pp. 41–48 (2006)

4. Argyriou, A., Evgeniou, T., Pontil, M.: Convex multi-task feature learning. Machine Learning **73**(3), 243–272 (2008)
5. Caruana, R.: Multitask learning. Machine Learning **28**(1), 41–75 (1997)
6. Dalal, N., Triggs, B.: Histogram of Oriented Gradients for Human Detection. Proc. CVPR. **2**, 886–893 (2005)
7. Daumé III, H.: Frustratingly easy domain adaptation. CoRR (2009)
8. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results (2007). http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html
9. Evgeniou, T., Micchelli, C.A., Pontil, M.: Learning multiple tasks with kernel methods. Journal of Machine Learning Research **6**, 615–637 (2005)
10. Evgeniou, T., Pontil, M.: Regularized multi-task learning. In: KDD 2004: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 109–117. ACM, New York (2004)
11. Felzenszwalb, P., Mcallester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: Proc. CVPR (2008)
12. Fornoni, M., Caputo, B., Orabona, F.: Multiclass latent locally linear support vector machines. In: ACML, pp. 229–244 (2013)
13. Fu, Z., Robles-Kelly, A., Zhou, J.: Mixing linear svms for nonlinear classification. IEEE Transactions on Neural Networks **21**(12), 1963–1975 (2010)
14. Jacob, L., Bach, F., Vert, J.: Clustered multi-task learning: A convex formulation. In: NIPS, pp. 745–752 (2008)
15. Johnson, S., Everingham, M.: Learning effective human pose estimation from inaccurate annotation. In: Proc. CVPR, pp. 1465–1472 (2011)
16. Kang, Z., Grauman, K., Sha, F.: Learning with whom to share in multi-task feature learning. In: ICML, pp. 521–528 (2011)
17. Kato, T., Kashima, H., Sugiyama, M., Asai, K.: Multi-task learning via conic programming. In: NIPS (2007)
18. Khosla, A., Zhou, T., Malisiewicz, T., Efros, A.A., Torralba, A.: Undoing the damage of dataset bias. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part I. LNCS, vol. 7572, pp. 158–171. Springer, Heidelberg (2012)
19. Ladicky, L., Torr, P.H.S.: Locally linear support vector machines. In: Proc. ICML, pp. 985–992 (2011)
20. Lampert, C.H., Blaschko, M.B.: Structured prediction by joint kernel support estimation. Machine Learning (2009)
21. Lee, S., Chatalbashev, V., Vickrey, D., Koller, D.: Learning a meta-level prior for feature relevance from multiple related tasks. In: ICML 2007: Proceedings of the 24th International Conference on Machine Learning, pp. 489–496. ACM, New York (2007)
22. Liu, J., Sun, J., Shum, H.: Paint selection. In: Proc. ACM SIGGRAPH (2009)
23. Malisiewicz, T., Gupta, A., Efros, A.A.: Ensemble of exemplar-SVMs for object detection and beyond. In: Proc. ICCV (2011)
24. Obozinski, G., Taskar, B., Jordan, M.I.: Joint covariate selection and joint subspace selection for multiple classification problems. Statistics and Computing **20**(2), 231–252 (2010)
25. Parameswaran, S., Weinberger, K.Q.: Large margin multi-task metric learning. In: NIPS, pp. 1867–1875 (2010)

26. Thrun, S.: Learning to learn: Introduction. Kluwer Academic Publishers (1996)
27. Yu, K., Tresp, V., Schwaighofer, A.: Learning gaussian processes from multiple tasks. In: ICML, pp. 1012–1019 (2005)
28. Zhang, Y., Yeung, D.: A convex formulation for learning task relationships in multi-task learning. CoRR abs/1203.3536 (2012)
29. Zhou, J., Chen, J., Ye, J.: Clustered multi-task learning via alternating structure optimization. In: NIPS, pp. 702–710 (2011)
30. Zhu, X., Vondrick, C., Ramanan, D., Fowlkes, C.: Do we need more training data or better models for object detection? In: Proc. BMVC, pp. 445–458 (2012)