Proc. IEEE Int. Conf. on Image Processing (ICIP 2016), Phoenix, AZ, USA, Sept. 2016. DOI: 10.1109/ICIP.2016.7532344

ONLINE MULTI-TASK LEARNING FOR SEMANTIC CONCEPT DETECTION IN VIDEO

Foteini Markatopoulou[1,2] *Vasileios Mezaris*[1] *Ioannis Patras*[2]

[1] Information Technologies Institute (ITI), CERTH, Thermi 57001, Greece
[2]Queen Mary University of London, Mile end Campus, UK, E14NS
{markatopoulou, bmezaris}@iti.gr i.patras@qmul.ac.uk

ABSTRACT

In this paper we propose an online multi-task learning algorithm for video concept detection. In particular, we extend the Efficient Lifelong Learning Algorithm (ELLA) in the following ways: a) we solve the objective function of ELLA using quadratic programming instead of solving the Lasso problem, b) we add a new label-based constraint that considers concept correlations, c) we use linear SVMs as base learners instead of logistic regression. Experimental results show improvement over both the single-task learning methods typically used in this problem and the original ELLA algorithm.

Index Terms— Concept detection, multi-task learning, video

1. INTRODUCTION

Semantic concept detection in video refers to the task of assigning one or more semantic concepts to video fragments (e.g., video keyframes) based on a predefined concept list (e.g., "car", "running") [1]. The typical process for this is that the video is initially segmented into meaningful fragments, called shots; each shot may be represented by one or more characteristic keyframes; and, several features (e.g., DCNN-based, hand-crafted) are extracted from either the keyframes or the entire shot. Then, for each target concept one supervised learning classifier is trained to solve the binary classification problem (i.e, decide on the presence or absence of the concept) [1]. Independently training concept detectors is a single-task learning (STL) process, where each task involves recognizing one concept. STL ignores the fact that groups of concepts can be related. Multi-task learning (MTL) refers to those methods that learn many tasks together at the same time, using a shared representation, which can result in better generalization performance. Traditional MTL is computationally expensive; once a new task arrives the shared representation should be learned from scratch. Lifelong or online MTL is new family of approaches that focuses on efficiently learning new concepts by building upon previous knowledge [2].

In this work we develop a lifelong MTL algorithm for video concept detection, referring to it as Efficient Lifelong Learning Algorithm with Label Constraint (ELLA_LC). Specifically, we extend the existing ELLA [2] in the following ways: a) we solve the objective function of ELLA using quadratic programming, b) we add a new label-based constraint that considers concept correlations and c) we instantiate ELLA with both linear Support Vector Machines (SVM) and Logistic regression (LR). We evaluate ELLA_LC on the TRECVID 2013 semantic indexing (SIN) task dataset on 38 different semantic concepts [3]. Our results show that the proposed algorithm leads to better concept-based video retrieval than existing state-of-the-art concept detection methods.

This work was supported by the EU's Horizon 2020 program under grant agreement H2020-687786 InVID.

2. RELATED WORK

A recent trend in video concept detection is to learn features directly from the raw keyframe pixels using deep convolutional neural networks (DCNNs). Most DCNN-based approaches perform the final class label prediction directly, using a softmax layer [4, 5]. Other approaches follow a STL process to train classifiers, e.g., SVMs, with features extracted from one or more layers of DCNNs [6]. More elaborate methods that consider the relations between concepts have also been proposed, divided in two main categories: i) methods that model the label relations [1, 7, 8, 9], i.e., the relations between concepts within a video shot, and ii) methods that model the task relations. A drawback of the methods in the first category is that they are affected by the quality of the ground-truth annotation (e.g., failing when there is only partially-annotated data). Furthermore, they ignore the fact that concept models, except for label relations, can be related through different structures captured either from the feature representation or the task parameters, i.e, the parameters of the binary classifier learned from the training data. In these two cases MTL methods that exploit task relations can perform better.

MTL methods learn the relations between many tasks together at the same time. The main difference of MTL methods is the way they define task relatedness. Some methods identify shared features between different task and use regularization to model task relatedness [10, 11, 12]. Other methods identify a shared subspace over the task parameters [13, 14, 15]. The methods above make the strong assumption that all tasks are related; some newer methods consider the fact that some tasks may be unrelated. For example, the clustered MTL algorithm (CMTL) [16] uses a clustering approach to assign on the same cluster parameters of tasks that lie nearby in terms of their L2 distance. Adaptive MTL (AMTL) [17] decomposes the task parameters into a low-rank structure that captures task relations, and a group-sparse structure that detects outlier tasks. The GO-MTL algorithm [18] uses a dictionary based method that allows two tasks from different groups to overlap by having one or more basis in common. In general, MTL methods have high computational cost, because the shared representation for all tasks is learned at once. Online or lifelong MTL, in contrast, can learn consecutive tasks that arrive in a sequence. The ELLA algorithm [2] is the online version of GO-MTL [18], presenting similar performance but being three orders of magnitude faster in training. In this work ELLA is used as the starting point for devising an online MTL algorithm suitable for video concept detection.

Proc. IEEE Int. Conf. on Image Processing (ICIP 2016), Phoenix, AZ, USA, Sept. 2016. DOI: 10.1109/ICIP.2016.7532344

3. OUR APPROACH

3.1. Problem formulation

A video concept detection system needs to learn a number of supervised learning tasks, one for each target concept. Each task t is associated with one concept detector (task model) $C^{(t)} : \mathbb{R}^d \mapsto \{\pm 1\}$, and the training set available for this concept $D^{(t)} = (\boldsymbol{x}_i^{(t)}, y_i^{(t)})_{i=1}^{n_t}$, where $\boldsymbol{x}_i^{(t)} \in \mathbb{R}^d$, $y_i^{(t)} \in \{\pm 1\}$. A concept detector can be defined as $C^{(t)}(\boldsymbol{x}^{(t)}) = C(\boldsymbol{x}^{(t)}; \boldsymbol{w}^{(t)})$, where $\boldsymbol{w}^{(t)} \in \mathbb{R}^d$ is the task parameter vector. A concept detection system should be easily extended with new tasks even if the total number of tasks $T_{\max}$ is not available from the start. Furthermore, task parameters of related concepts may share similar knowledge but also concept correlations obtained by the ground-truth annotation provide another source of information regarding the relations between tasks. In this work, considering all the above we propose the ELLA_LC algorithm for video concept detection.

3.2. ELLA with label constraints: ELLA_LC

ELLA_LC is an extension of the ELLA algorithm proposed by Eaton and Ruvolo [2]. Specifically, we add a new label-based constraint on ELLA's model in order to incorporate statistical information of pairwise correlations between concepts that we can acquire from the ground-truth annotation. ELLA_LC uses a knowledge shared basis $\boldsymbol{L} \in \mathbb{R}^{d \times k}$ for all task models, where the columns of $\boldsymbol{L}$ correspond to the parameter vectors of k latent tasks. We model the parameter vector $\boldsymbol{w}^{(t)}$ of observed task t as $\boldsymbol{w}^{(t)} = \boldsymbol{L}\boldsymbol{s}^{(t)}$, where $\boldsymbol{s}^{(t)} \in \mathbb{R}^k$ is a task-specific weight vector that contains the coefficients of the linear combination. Each linear combination is assumed to be sparse in $\boldsymbol{L}$; in this way we assume that there exist a small number of latent basis tasks and the task parameter vector of every observed task $\boldsymbol{w}^{(t)}$ is a linear combination of them. The overlap in the sparsity patterns of any two tasks controls the amount of sharing between them. ELLA_LC builds a concept detection system that i) updates the shared basis $\boldsymbol{L}$ when a new concept arrives without building all the previous task models from scratch; and ii) incorporates the label correlations of the new concept with all of the previously learned concepts in order to improve the learning of the task-specific weight vector regarding the new concept and the shared basis $\boldsymbol{L}$. The above problem can be formulated by the following objective function:

$$\min_{(\mathbf{L},\boldsymbol{s}^{(t)})} \frac{1}{T}\sum_{t=1}^{T}\left\{\frac{1}{n_t}\sum_{i=1}^{n_t}\mathcal{L}\Big(C(\boldsymbol{x}_i^{(t)};\boldsymbol{L}\boldsymbol{s}^{(t)}), y_i^{(t)}\Big) + \mu\left\|\boldsymbol{s}^{(t)}\right\|_1 \right. \quad (1)$$

$$\left. +\beta\Bigg(\sum_{\substack{t'=1\\ t'\neq t}}^{T}\frac{1}{T\text{-}1}|r_{t,t'}|\left\|\boldsymbol{L}(\boldsymbol{s}^{(t)} - \text{sign}(r_{t,t'})\boldsymbol{s}^{(t')})\right\|^2\Bigg)\right\} + \lambda\left\|\boldsymbol{L}\right\|_F^2,$$

where $\mathcal{L}$ refers to the loss function that is used for learning the task specific parameter vector $\boldsymbol{w}^{(t)}$; $\boldsymbol{w}^{(t)}$ is obtained by building a classifier only using the training data available for task t: $\min_{(\boldsymbol{w}^{(t)})} \frac{1}{n_t}\sum_{i=1}^{n_t}\mathcal{L}\Big(C(\boldsymbol{x}_i^{(t)};\boldsymbol{w}^{(t)}, y_i^{(t)})\Big)$. T is the number of tasks that have been learned so far. $r_{t,t'}$ is the ϕ-correlation coefficient of the concepts learned regarding the tasks t and t', calculated from the ground-truth annotation of the training set.

This is similar to the objective function of ELLA [2] with the addition of the extra term: $|r_{t,t'}|\left\|\boldsymbol{L}(\boldsymbol{s}^{(t)} - \text{sign}(r_{t,t'})\boldsymbol{s}^{(t')})\right\|^2$ that incorporates the label correlations. Specifically, we aim to model highly correlated concepts (positive or negative). We assume that if two concepts are positively correlated, then the underlying task parameters should be similar; on the other hand, if two concepts are negatively correlated, then the task parameters should be opposite. The similarity measure suitable to capture both positive and negative correlations that was selected in this study is the ϕ-correlation coefficient. To model this assumption we use the above constraint in the objective function of Eq. (1). The larger the correlation between two concepts (positive or negative), the higher the imposed constraint. On the other hand, if two concepts are not correlated, using the above function will not impose any constraint. This constraint is applicable to linear classifiers, where positive correlated concepts are forced to return similar responses, while negative correlated concepts are forced to return opposite responses.

Algorithm 1 ELLA_LC(k,d,λ,μ,β)

while existMoreTrainingData() **do**
$T \leftarrow T$+1, $t = T$
1. $(\boldsymbol{w}^{(t)}, \boldsymbol{H}^{(t)}) \leftarrow$ base_learner$(\boldsymbol{X}_{\text{new}}^{(t)}, \boldsymbol{y}_{\text{new}}^{(t)})$
2. $\boldsymbol{s}^{(t)} \leftarrow$ Eq. (3)
3. $\boldsymbol{A} \leftarrow \boldsymbol{A} + \Big((\boldsymbol{s}^{(t)}\boldsymbol{s}^{(t)\mathsf{T}}) \otimes \boldsymbol{H}^{(t)} + \beta\Big(\sum_{\substack{t'=1\\ t'\neq t}}^{T}\frac{1}{T\text{-}1}|r_{t,t'}|[\boldsymbol{\phi}_{t,t'}\boldsymbol{\phi}_{t,t'}^{\mathsf{T}} \otimes \boldsymbol{I}]\Big)\Big)$
4. $\boldsymbol{b} \leftarrow \boldsymbol{b} + \text{vec}(\boldsymbol{s}^{(t)\mathsf{T}} \otimes (\boldsymbol{w}^{(t)\mathsf{T}}\boldsymbol{H}^{(t)}))$
5. $\boldsymbol{L} \leftarrow \text{mat}((\frac{1}{T}\boldsymbol{A} + \lambda\boldsymbol{I}_{k\times d, k\times d})^{-1}\frac{1}{T}\boldsymbol{b})$
end while

3.3. Problem solution

Equation (1) is not jointly convex in $\boldsymbol{L}$ and $\boldsymbol{s}^{(t)}$, so ELLA [2] approximates them using two simplifications that we also follow here: i) to eliminate the explicit dependence on all previous training data through the inner summation, we approximate Eq. (1) using the second-order Taylor expansion of $\frac{1}{n_t}\sum_{i=1}^{n_t}\mathcal{L}\Big(C(\boldsymbol{x}_i^{(t)};\boldsymbol{w}^{(t)}), y_i^{(t)}\Big)$ around $\boldsymbol{w}^{(t)}$. ii) We compute each $\boldsymbol{s}^{(t)}$ only when training data for task t are available and do not update it when new tasks arrive. These give the following objective function that approximates Eq. (1):

$$\min_{(\mathbf{L},\boldsymbol{s}^{(t)})} \frac{1}{T}\sum_{t=1}^{T}\left\{\left\|\boldsymbol{w}^{(t)} - \boldsymbol{L}\boldsymbol{s}^{(t)}\right\|_{\boldsymbol{H}^{(t)}}^2 + \mu\left\|\boldsymbol{s}^{(t)}\right\|_1 \right. \quad (2)$$

$$\left. +\beta\Bigg(\sum_{\substack{t'=1\\ t'\neq t}}^{T}\frac{1}{T\text{-}1}|r_{t,t'}|\left\|\boldsymbol{L}(\boldsymbol{s}^{(t)} - \text{sign}(r_{t,t'})\boldsymbol{s}^{(t')})\right\|^2\Bigg)\right\} + \lambda\left\|\boldsymbol{L}\right\|_F^2,$$

where $\boldsymbol{H}^{(t)}$ is the Hessian of the loss function $\mathcal{L}$ evaluated on $\boldsymbol{w}^{(t)}$. To optimize this objective function, consequently to update our model, we perform three steps: i) we compute the ϕ-correlation coefficient of the concept learned in task t with all the previously learned concepts; ii) we compute the task-specific weight vector $\boldsymbol{s}^{(t)}$ and iii) we update the shared basis $\boldsymbol{L}$. Below we describe each of these steps in more details.

To compute $\boldsymbol{s}^{(t)}$ we solve Eq. (2) for $\boldsymbol{s}^{(t)}$, when the task t arrives

Proc. IEEE Int. Conf. on Image Processing (ICIP 2016), Phoenix, AZ, USA, Sept. 2016. DOI: 10.1109/ICIP.2016.7532344

(i.e., $t = T$):

$$\min_{\boldsymbol{s}^{(t)}} \left\{ \left\| \boldsymbol{w}^{(t)} - \boldsymbol{L}\boldsymbol{s}^{(t)} \right\|^2_{\boldsymbol{H}^{(t)}} + \mu \left\| \boldsymbol{s}^{(t)} \right\|_1 + \beta \left(\sum_{\substack{t'=1 \\ t' \neq t}}^{t} \frac{1}{t\text{-}1} |r_{t,t'}| \left\| \boldsymbol{L}(\boldsymbol{s}^{(t)} - \text{sign}(r_{t,t'})\boldsymbol{s}^{(t')}) \right\|^2 \right) \right\} \quad (3)$$

By expanding the above we arrive to the following problem that can be solved using quadratic programming:

$$\min_{\boldsymbol{s}^{(t)}} \left\{ \boldsymbol{s}^{(t)^\mathsf{T}} \left[\boldsymbol{L}^\mathsf{T} \boldsymbol{H}^{(t)} \boldsymbol{L} + \beta \left(\sum_{\substack{t'=1 \\ t' \neq t}}^{t} \frac{1}{t\text{-}1} |r_{t,t'}| \boldsymbol{L}^\mathsf{T} \boldsymbol{L} \right) \right] \boldsymbol{s}^{(t)} \right. \quad (4)$$
$$\left. -2 \left[\boldsymbol{w}^{(t)^\mathsf{T}} \boldsymbol{H}^{(t)} \boldsymbol{L} + \frac{\beta}{2} \left(\sum_{\substack{t'=1 \\ t' \neq t}}^{t} \frac{1}{t\text{-}1} r_{t,t'} \boldsymbol{s}^{(t')^\mathsf{T}} \boldsymbol{L}^\mathsf{T} \boldsymbol{L} \right) - \frac{\mu}{2} \boldsymbol{I} \right] \boldsymbol{s}^{(t)} \right\}$$

Similarly, to update $\boldsymbol{L}$, we solve Eq. (2) for $\boldsymbol{L}$, which equals to:

$$\min_{\mathbf{L}} \frac{1}{T} \sum_{t=1}^{T} \left\{ \left\| \boldsymbol{w}^{(t)} - \boldsymbol{L}\boldsymbol{s}^{(t)} \right\|^2_{H} + \beta \left(\sum_{\substack{t'=1 \\ t' \neq t}}^{T} \frac{1}{T\text{-}1} |r_{t,t'}| \left\| \boldsymbol{L}\boldsymbol{\phi}_{t,t'} \right\|^2 \right) \right\} + \lambda \left\| \boldsymbol{L} \right\|^2_F, \quad (5)$$

where $\boldsymbol{\phi}_{t,t'} = \boldsymbol{s}^{(t)} - \text{sign}(r_{t,t'})\boldsymbol{s}^{(t')}$. Then, we null the gradient of Eq. (5) and solve for $\boldsymbol{L}$. This gives a column-wise vectorization of $\boldsymbol{L}$ as $\boldsymbol{A}^{-1}\boldsymbol{b}$ where:

$$\boldsymbol{A} = \frac{1}{T} \sum_{t=1}^{T} \left\{ (\boldsymbol{s}^{(t)} \boldsymbol{s}^{(t)^\mathsf{T}}) \otimes \boldsymbol{H}^{(t)} + \beta \left(\sum_{\substack{t'=1 \\ t' \neq t}}^{T} \frac{1}{T\text{-}1} |r_{t,t'}| [\boldsymbol{\phi}_{t,t'} \boldsymbol{\phi}_{t,t'}^\mathsf{T} \otimes \boldsymbol{I}] \right) \right\} + \lambda \boldsymbol{I}, \text{ and} \quad (6)$$

$$\boldsymbol{b} = \frac{1}{T} \sum_{t=1}^{T} vec(\boldsymbol{s}^{(t)^\mathsf{T}} \otimes (\boldsymbol{w}^{(t)^\mathsf{T}} \boldsymbol{H}^{(t)}))$$

Algorithm 1 summarizes the steps that ELLA_LC performs for updating the above parameters and learning consecutive tasks. In each iteration, ELLA_LC receives training data $(\boldsymbol{X}^{(t)}_{new}, \boldsymbol{y}^{(t)}_{new})$ for a task t. If this is the first time that task t appears, then ELLA_LC computes the model parameters $\boldsymbol{w}^{(t)}$ and Hessian $\boldsymbol{H}^{(t)}$ from only this training data for task t using a base learner (Alg. 1: Step 1). This step depends on the base learning algorithm; Eaton and Ruvolo [2] provide details for learning linear and logistic regression models using ELLA. In the next section we show how the hinge loss can be used instead. We compute the ϕ-correlation coefficient of the concept learned in task t with all the previously learned concepts. Subsequently, the correlation information and the shared basis $\boldsymbol{L}$ are used to compute the task-specific weight vector $\boldsymbol{s}^{(t)}$ (Alg. 1: Step 2). Finally, ELLA_LC updates the basis $\boldsymbol{L}$ to incorporate new knowledge via an incremental update that considers Eq. (6) (Alg. 1: Steps 3-5). If additional training data for a previously learned task is provided, the algorithm can be extended, similarly to [2], in order to concatenate the new data with the past data and then update the vector $\boldsymbol{s}^{(t)}$ and the shared basis $\boldsymbol{L}$.

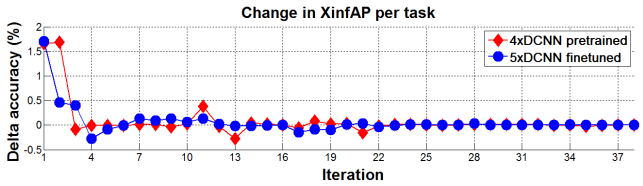

Fig. 1. Change in XinfAP for each task between the iteration that the task was first learned and the last iteration (where all tasks had been learned), divided by the position of the task in the task sequence.

3.4. L2-loss (squared hinge loss) support vector classification

In this section we will show how to use SVMs as the base learner for ELLA_LC (Algorithm 1: Step 1). In this case, $\boldsymbol{y}^{(t)} \in \{\pm 1\}^{n_t}$ and $C(\boldsymbol{x}^{(t)}; \boldsymbol{w}^{(t)}) = \boldsymbol{w}^{(t)^\mathsf{T}} \boldsymbol{x}^{(t)}$; $\mathcal{L}$ is the L2-loss (squared hinge loss) function: $\max(0, 1 - \boldsymbol{y}^{(t)} \boldsymbol{w}^{(t)^\mathsf{T}} \boldsymbol{x}^{(t)})^2$. Firstly, we use an algorithm for L2-loss SVM to compute the task parameter vector $\boldsymbol{w}^{(t)}$. Then, the Hessian $\boldsymbol{H}^{(t)}$ of the L2-loss function evaluated on $\boldsymbol{w}^{(t)}$ is given as follows:

$$\boldsymbol{H}^{(t)} = \begin{cases} \frac{1}{n_t} \sum_{i=1}^{n_t} \boldsymbol{x}_i^{(t)} \boldsymbol{x}_i^{(t)^\mathsf{T}}, & \text{if } y_i \boldsymbol{w}^{(t)^\mathsf{T}} \boldsymbol{x}_i^{(t)} < 1 \\ 0, & \text{otherwise} \end{cases}$$

4. EXPERIMENTS

4.1. Dataset and experimental setup

Our experiments were performed on the TRECVID 2013 SIN dataset [3], which consists of approximately 800 and 200 hours of internet archive videos for training and testing, respectively. We evaluated our system on the test set using the 38 concepts that were evaluated as part of the TRECVID 2013 SIN task [3]. The video indexing problem was examined; that is, given a concept, we measure how well the top retrieved video shots for this concept truly relate to it. We analyse our results in terms of mean extended inferred average precision (MXinfAP) [19], which is an approximation of the mean average precision suitable for the partial ground-truth that accompanies the TRECVID dataset [3]. We used features from four different DCNNs that were trained on ImageNet data [20]: i) The 8-layer CaffeNet [5], ii) the 16-layer deep ConvNet [4], iii) the 22-layer GoogLeNet [21], iv) a DCNN that we trained according to the 22-layer GoogLeNet architecture on the ImageNet "fall" 2011 dataset for 5055 categories. We will refer to these networks as CaffeNet1K, ConvNet1K, GNET1k, GNET5k, respectively. We applied each of these networks on the TRECVID keyframes and we used as a feature the network's direct output, that corresponds to the final class label prediction for 1000 ImageNet categories for CaffeNet1k, ConvNet1k and GNET1k and 5055 categories for GNET5k. All the feature vectors were finally reduced to 400 dimensions using principal components analysis (PCA), as this was shown to improve the performance for all the methods in our experiments by around 2% (in terms of MXinfAP). To improve the results, we also experimented with fine-tuning (FT) the above methods on various subsets of the 346 TRECVID SIN concepts according to [22]. This resulted in 5 FT networks that differ in the number and dimension of the extension layers and in the number of output categories. Subsequently, we again applied these FT networks on the TRECVID keyframes. To train our base classifiers, for each concept, a training set was

Proc. IEEE Int. Conf. on Image Processing (ICIP 2016), Phoenix, AZ, USA, Sept. 2016. DOI: 10.1109/ICIP.2016.7532344

Table 1. MXinfAP for 38 concepts using different features or combinations of them: i) using networks that have been trained on ImageNet data, ii) using ImageNet networks that have been fine-tuned on the TRECVID SIN 2013 training set. After the "-" symbol we indicate the number of concepts for which the network has been finetuned. For 4xDCNN and 5xDCNN FT the direct output of the four ImageNet and five FT networks, respectively, was fused, in terms of arithmetic mean, and served as input to the learning algorithms. The * symbol indicates that the difference in MXinfAP between the denoted method and the best-performing method in the same row of the table is not statistically significant.

			Single-task learning			**Joint concept learning**				**Proposed multi-task learning**			
R#	**Features**	**Direct output**	**LR**	**LSVM**	**KSVM**	**LP [9]**	**AMTL [17]**	**CMTL [16]**	**ELLA [2]**	**ELLA_QP LR**	**ELLA_QP LSVM**	**ELLA_LC LR**	**ELLA_LC LSVM**
	(i) Using the output of ImageNet-based networks as features												
1	CaffeNet1k	-	13.00	14.20	12.81	11.77	12.90	11.56	13.14	13.99	16.27 *	14.28	**16.36**
2	ConvNet1k	-	17.58	19.29	15.62	1608	17.58	16.09	17.88	18.45	21.02 *	18.94	**21.10**
3	GNET1k	-	16.10	17.73	14.17	15.00	16.34	14.43	15.79	17.07	19.86 *	17.48	**19.98**
4	GNET5k	-	20.89	22.68	20.73	20.54	21.01	19.99	15.65	21.88	24.05 *	22.16	**24.14**
5	**4xDCNN**	-	21.77	24.29	22.64	19.58	22.96	21.42	21.17	23.66	25.97 *	24.18	**26.10**
	(ii) Using the output of networks finetuned on different subsets of the TRECVID SIN 2013 training set as features												
6	CaffeNet1k-345	20.29	22.21	24.16	23.00	21.29	24.22	24.03	16.63	23.09	25.47 *	23.51	**25.88**
7	GNET1k-60	19.77	24.51	24.30	23.07	25.06 *	22.56	22.25	23.71	24.56	**26.05**	24.51	25.90 *
8	GNET1k-60	19.90	24.71	24.78	22.90	25.20 *	23.87	22.87	24.57	24.69	**26.24**	24.52	**26.24**
9	GNET1k-323	23.97	26.67	28.65	27.79	27.22	28.67	28.09	25.75	27.56	29.86	28.19	**30.23**
10	GNET5k-323	22.78	27.13	29.32	28.53	28.21	29.47	29.27	27.15	28.61	30.80 *	28.90	**31.01**
11	**5xDCNN FT**	25.35	28.56	30.60	29.93	30.27	30.94	30.15	28.19	29.89	31.82 *	30.32	**32.10**

assembled that included all positive annotated training examples for the given concept, and negatives to a maximum of 15:1 ratio.

We instantiated the proposed ELLA_LC with two base classifiers: LR and Linear SVM (LSVM). We performed comparisons with the following methods: i) STL using a) LR, b) LSVM and c) kernel SVM with radial kernel (KSVM). ii) The label powerset (LP) multi-label learning algorithm, that has been used in [9] to model the label relations. iii) AMTL [17], and iv) CMTL [16], two batch MTL methods, v) ELLA [2], an online MTL approach. We selected all the parameter values for these methods based on the training data alone. The value of k was set to 38, regularization parameters λ and μ in Eq. (1) were kept fixed at exp(-10) and exp(-3), respectively, for all the online MTL methods, in all experiments. These parameters are expected to depend on the dimensionality of the feature space and the number of examples, and according to preliminary experiments seem to work well for the employed features. The parameter value of β for ELLA_LC was selected from the set {exp(-2), exp(-1),1, exp(1), exp(2)}. The ordering of the tasks was fixed and the same for all the online MTL algorithms, and each task was presented sequentially. AMTL and CMTL learned each new task from a single batch of data that contained all training instances of that task. For implementing the above techniques, the LibLINEAR library [23] was used as the source of learning LR and LSVM models. For the LSVM we used the L2 regularized L2-loss SVC and solved it in the primal form. The LibSVM [24] and the MULAN [25] libraries were used as the source for the KSVM and LP algorithms [9], respectively. The MALSAR library [26] was used for learning the CMTL [16] and AMTL [17].

4.2. Experimental results

Table 1 presents the results of our experiments in terms of MXinfAP. ELLA_QP is an intermediate version of the proposed ELLA_LC that solves the objective function of ELLA [2] with respect to $\boldsymbol{s}^{(t)}$ using quadratic programming (QP), instead of solving the Lasso problem [2], but does not use the label constraint of ELLA_LC. Starting from the upper part of Table 1, which refers to features extracted from ImageNet DCNN networks, we can see that the proposed ELLA_QP and ELLA_LC perform better than the STL alternatives both when LR and when LSVM is used as the base learner. In addition, solving Eq. (2) with QP (ELLA_QP) outperforms the original ELLA [2]. Adding also the label constraint (ELLA_LC) further improves the ELLA_QP method for all of the feature types. The proposed ELLA_LC with LSVM is the best performing method, reaching a MXinfAP of 26.10%. Similar conclusions can be reached if we look in the lower part of Table 1, where features extracted from the FT networks are used. Furthermore, we observe that fine-tuning is a procedure that significantly improves the retrieval accuracy of all the compared methods, with ELLA_LC reaching once again the highest performance (MXinfAP equal to 32.10%). To investigate the statistical significance of the difference of each method from the best performing method we used a paired t-test as suggested by [27]; in Table 1, the absence of * suggests statistical significance. We found that differences between the proposed ELLA_LC and all the other methods are significant (at 5% significance level) except for two runs based on LP (Table 1: R7-8) and all but one of the ELLA_QP LSVM runs (Table 1: R9). Where the latter is also proposed in the paper. We note that although the differences between ELLA_QP and ELLA_LC are in most cases not significant, ELLA_LC exhibits better results consistently across the different runs (R1-11, except R7,8).

ELLA_LC updates the value of $\boldsymbol{s}^{(t)}$ only when it receives new training data for task t. This online characteristic of ELLA makes it faster than the batch MTL methods. At the same time, the update of $\boldsymbol{L}$ benefits the previously learned tasks by introducing new knowledge acquired from the last learned task. To assess whether the latter occurred, in Fig. 1, for each task we computed the change in XinfAP from when the task was first learned and the task's XinfAP in the last iteration (where all tasks had been learned). We normalize this quantity by the task's position in the task sequence. We can see that reverse transfer occurred, i.e., a positive change in accuracy for a task indicates this, mainly for the tasks that were learned early. As far as the pool of tasks increases early tasks get new knowledge from many more tasks, which explains why the benefit is bigger for them.

5. CONCLUSIONS

In this work we presented an online MTL method for video concept detection. Extensive experiments reveal the usefulness of learning the relations between many task models (one per concept) in combination with the concept correlations that can be captured from the ground-truth annotation.

6. REFERENCES

[1] C. G. M. Snoek and M. Worring, "Concept-Based Video Retrieval," *Foundations and Trends in Information Retrieval*, vol. 2, no. 4, pp. 215–322, 2009.

[2] E. Eaton and P. L. Ruvolo, "Ella: An efficient lifelong learning algorithm," in *Proc. of the 30th Int. Conf. on Machine Learning (ICML-13)*, S. Dasgupta and D.Mcallester, Eds. 2013, vol. 28, pp. 507–515, JMLR Workshop.

[3] P. Over et al., "Trecvid 2013 – an overview of the goals, tasks, data, evaluation mechanisms and metrics," in *Proc. of TRECVID 2013*. NIST, USA, 2013.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR, abs/1409.1556*, 2014.

[5] A. Krizhevsky, S. Ilya, and G.E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS 2012)*, pp. 1097–1105. Curran Associates, Inc., 2012.

[6] F. Markatopoulou, V. Mezaris, and I. Patras, "Cascade of classifiers based on binary, non-binary and deep convolutional network descriptors for video concept detection," in *Proc. of the IEEE Int. Conf. on Image Processing (ICIP 2015)*, 2015, pp. 1786–1790.

[7] Guo-Jun Qi et al., "Correlative multi-label video annotation," in *Proc. of the 15th Int. Conf. on Multimedia*, NY, 2007, pp. 17–26, ACM.

[8] M. Wang, X. Zhou, and T.-S. Chua, "Automatic image annotation via local multi-label classification," in *Proc. of the Int. Conf. on Content-based image and video retrieval (CIVR '08)*, NY, 2008, pp. 17–26, ACM.

[9] F. Markatopoulou, V. Mezaris, N. Pittaras, and I. Patras, "Local features and a two-layer stacking architecture for semantic concept detection in video," *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 2, pp. 193–204, 2015.

[10] A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-task feature learning," in *Advances in Neural Information Processing Systems (NIPS 2007)*. 2007, MIT Press.

[11] G. Obozinski and B. Taskar, "Multi-task feature selection," in *Proc. of the 23rd Int. Conf. on Machine Learning (ICML 2006). Workshop of Structural Knowledge Transfer for Machine Learning*, Pittsburgh, Pennsylvania, 2006.

[12] H.S. Mousavi, U. Srinivas, V. Monga, Yuanming Suo, Minh Dao, and T.D. Tran, "Multi-task image classification via collaborative, hierarchical spike-and-slab priors," in *Proc. of the IEEE Int. Conf. on Image Processing (ICIP 2014)*, 2014, pp. 4236–4240.

[13] T. Evgeniou and M. Pontil, "Regularized multi–task learning," in *Proc. of the 10th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD '04)*, NY, USA, 2004, pp. 109–117, ACM.

[14] H. Daumé, III, "Bayesian multitask learning with latent hierarchies," in *Proc. of the 25th Conf. on Uncertainty in Artificial Intelligence (UAI '09)*, Arlington, Virginia, US, 2009, pp. 135–142, AUAI Press.

[15] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Machine Learning*, vol. 73, no. 3, pp. 243–272, 2008.

[16] J. Zhou, J. Chen, and J. Ye, "Clustered multi-task learning via alternating structure optimization," *Advances in Neural Information Processing Systems (NIPS 2011)*, 2011.

[17] G. Sun, Y. Chen, X. Liu, and E. Wu, "Adaptive multi-task learning for fine-grained categorization," in *Proc. of the IEEE Int. Conf. on Image Processing (ICIP 2015)*, 2015, pp. 996–1000.

[18] A. Kumar and H. Daume, "Learning task grouping and overlap in multi-task learning," in *Proc. of the 29th Int. Conf. on Machine Learning (ICML-12)*, John Langford and Joelle Pineau, Eds., NY, USA, 2012, pp. 1383–1390, ACM.

[19] E. Yilmaz, E. Kanoulas, and J. A. Aslam, "A simple and efficient sampling method for estimating ap and ndcg," in *31st ACM SIGIR Int. Conf. on Research and Development in Information Retrieval*, USA, 2008, pp. 603–610, ACM.

[20] O. Russakovsky, J. Deng, and H. Su et al., "ImageNet Large Scale Visual Recognition Challenge," *Int. Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[21] C. Szegedy et al., "Going deeper with convolutions," in *CVPR 2015*, 2015.

[22] F. Markatopoulou, A. Ioannidou, and C. Tzelepis et al., "ITI-CERTH participation to TRECVID 2015," in *Proc. of TRECVID 2015*. NIST, USA, 2015.

[23] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[24] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[25] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas, "Mulan: A java library for multi-label learning," *Journal of Machine Learning Research*, vol. 12, pp. 2411–2414, 2011.

[26] J. Zhou, J. Chen, and J. Ye, *MALSAR: Multi-task Learning via Structural Regularization*, Arizona State University, 2011.

[27] Henk M. Blanken, Arjen P. de Vries, Henk Ernst Blok, and Ling Feng, *Multimedia Retrieval*, Springer Berlin Heidelberg, NY, 2005.