Multi-Task Transfer Methods to Improve One-Shot Learning for Multimedia Event Detection

Wang Yan
wyan@sfu.ca
Jordan Yap
jjyap@sfu.ca
Greg Mori
mori@cs.sfu.ca

School of Computing Science
Simon Fraser University
Burnaby, BC, CANADA

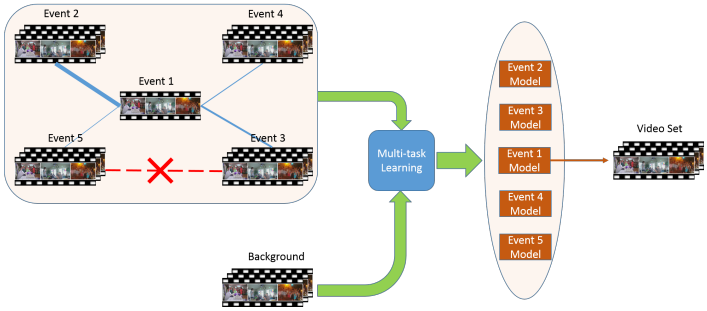

Figure 1: Overview of the proposed method – one-shot learning leveraging other categories. Given one example for an event of interest (Event 1), we implicitly infer the relevance between it and other events, and emphasise more on the most relevant ones in the multi-task learning. The learned classifier for the event of interest is applied to retrieve instances from a video test set.

This paper proposes a new multi-task learning method with implicit inter-task relevance estimation, and applies it to complex Internet video event detection, which is a challenging and important problem in practice, yet seldom has been addressed. In this paper, "detection" means to detect videos corresponding to the event of interest from a (large) video dataset, not to localize the event spatially or temporally in a video. In the problem definition, one positive and plenty of negative samples of one event are given as training data, and the goal is to return the videos of the same event from a large video dataset. In addition, we assume samples of other events are available.

Fig. 1 shows an overview of the proposed methods. The widths of the lines between the one-exemplar event and others represent the inter-event relevance, which is unknown a priori in our problem settings. However, the proposed method can implicity infer the relevance and utilize the most relevant event(s) more in multi-task learning, where the shared information from the relevant events helps to build a better model from the one exemplar. The proposed method does not assume the relevance between other events, as indicated by the red line. Although the learning algorithm outputs models of all input events, only that of the one-exemplar event is applied to detect videos of the event of interest from the video set.

Our method builds on the approach of graph-guided multi-task learning [1], which is described first. The training set $\{(\mathbf{x}_{ti}, y_{ti}) \in \mathbb{R}^D \times \{-1,+1\}, t = 1,2,\ldots,T, i = 1,2,\ldots,N_t\}$ is grouped into T related tasks, which are further organized as a graph $G =< V,E >$. The tasks correspond to the elements in the vertex set V, and the pairwise relevance between Task t and k are represented by the weight r_{tk} on edges $e_{tk} \in E$. The more relevant the two tasks are, the larger the edge weight is. The graph guided multi-task learning algorithm learns the corresponding T models jointly, by solving the optimization problem

$$\min_{\mathbf{W},\mathbf{b}} \sum_{t=1}^{T} \sum_{i=1}^{N_t} \text{Loss}(\mathbf{w}_t^T \mathbf{x}_{ti} + b_t, y_{ti}) + \lambda_1 \|\mathbf{W}\|_F^2 + \lambda_2 \Omega(\mathbf{W}), \quad (1)$$

where $\mathbf{w}_t \in \mathbb{R}^D$ and $b_t \in \mathbb{R}$ are the model weight vector and bias term of Task t, respectively, $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_T)$ is the matrix whose columns are model weight vectors, $\mathbf{b} = (b_1, b_2, \ldots, b_t)$, $\|\mathbf{W}\|_F^2 = \text{Trace}(\mathbf{W}^T\mathbf{W})$ is the squared Frobenius norm,

$$\Omega(\mathbf{W}) = \sum_{e_{tk} \in E, t<k} r_{tk} \|\mathbf{w}_t - \mathbf{w}_k\|_2^2 \quad (2)$$

is the graph-guided penalty term. For the significantly relevant tasks, their model weight vectors are forced to be similar due to the large edge weights, and the information could be transferred between relevant tasks. $\text{Loss}(\cdot,\cdot)$ is the loss function. In our work we use logistic loss

$$\text{Loss}(s,y) = \log(1 + \exp(-ys)), \quad (3)$$

which is smooth and leads to an easier optimization problem compared to the hinge loss.

In the one-shot learning setting for event detection, it is hard to learn a good model for the specific event of interest from the only one positive sample. However, due to the potential relevance between events, one can expect a better model by applying multi-task learning to the event with one positive sample and some other events. In the multi-task setting, each of these other events corresponds to one task.

Different external events may share different common "part" with the event of interest. Consider that "birthday party" as the event of interest, and it is relevant to "parade" since both have lots of people inside, and it may be also relevant to "preparing food" due to the food itself. However, there is little relevance between "parade" and "preparing food". Therefore, cluster-based multi-task learning may not be used to learning the event of interest, because not all external events relevant to the event of interest are relevant enough to each other to fit into one cluster. In contrast, graph-guided multi-task does not assume the clustered structure of the tasks, and it a better choice for this task.

Without losing the generality, we assume Event 1 is the event of interest and others are external event in the following. Given the graph in Fig. 1, the formulation in (1) is not directly applicable because the pairwise relevance is unknown. However, the min operation can be added to select the most relevant tasks automatically, and they are all equally weighted, i.e. using

$$\Omega(\mathbf{W}) = \sum_{t \in \mathcal{T}_K} \|\mathbf{w}_1 - \mathbf{w}_t\|_2^2 \quad (4)$$

instead of (2), where $\mathcal{T}_K$ is the set of indices corresponding to the minimum K elements in $\{\|\mathbf{w}_1 - \mathbf{w}_t\|_2^2\}_{t=2}^T$. The most relevant tasks and task models are jointly optimised in the training process. The minimum operation can be further replaced by softmin function to make the objective smooth, i.e.

$$\Omega(\mathbf{W}) = -\log \sum_{t=2}^{T} \exp(-\|\mathbf{w}_1 - \mathbf{w}_t\|_2^2). \quad (5)$$

This penalty focuses more on the smallest inter-model distance, which is slightly different from the former one with equally weighted K smallest distances. The experiments show that we can get good results with the smooth penalty. In addition to squared l_2 distance, one can also use the penalty term represented by correlations. The term still focuses more on most relevant tasks, but softmax is used in the representation, i.e.

$$\Omega(\mathbf{W}) = -\log \sum_{t=2}^{T} \exp(\mathbf{w}_1^T \mathbf{w}_t). \quad (6)$$

We use the first option in the following experiments. All penalties in this subsection make the objective non-convex, but one can still get good results empirically. The objective is optimized by Quasi-Newton Soft-Threshold (QNST) method [2].

[1] Theodoros Evgeniou, Charles A Micchelli, and Massimiliano Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.

[2] Mark Schmidt. *Graphical model structure learning with l_1-regularization*. PhD thesis, UNIVERSITY OF BRITISH COLUMBIA, 2010.