Computer Vision and Image Understanding 152 (2016) 79–87

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Video Classification via Weakly Supervised Sequence Modeling

Jingjing Liu [a,*], Chao Chen [b], Yan Zhu [a], Wei Liu [c], Dimitris N. Metaxas [a]

[a] Department of Computer Science, Rutgers, Piscataway Township, NJ 08854, USA
[b] Department of Computer Science, CUNY Queens College, Flushing, NY 11367, USA
[c] Didi Research, Beijing 100085, China

ARTICLE INFO

Article history:
Received 12 December 2014
Accepted 21 October 2015
Available online 10 November 2015

Keywords:
Video classification
Gesture
Action
Weakly supervised
Sequence modeling
Multiple-instance learning (MIL)
Conditional Random Fields (CRFs)

ABSTRACT

Traditional approaches for video classification treat the entire video clip as one data instance. They extract visual features from video frames which are then quantized (e.g., K-means) and pooled (e.g., average pooling) to produce a single feature vector. Such holistic representations of videos are further used as inputs of a classifier. Despite of efficiency, global and aggregate feature representation unavoidably brings in redundant and noisy information from background and unrelated video frames that sometimes overwhelms targeted visual patterns. Besides, temporal correlations between consecutive video frames are also ignored in both training and testing, which may be the key indicator of an action or event. To this end, we propose Weakly Supervised Sequence Modeling (WSSM), a novel framework that combines multiple-instance learning (MIL) and Conditional Random Field (CRF) model seamlessly. Our model takes each entire video as a *bag* and one video segment as an *instance*. In our framework, the salient local patterns for different video categories are explored by MIL, and intrinsic temporal dependencies between instances are explicitly exploited using the powerful chain CRF model. In the training stage, we design a novel conditional likelihood formulation which only requires annotation on videos. Such likelihood can be maximized using an alternating optimization method. The training algorithm is guaranteed to converge and is very efficient. In the testing stage, videos are classified by the learned CRF model. The proposed WSSM algorithm outperforms other MIL-based approaches in both accuracy and efficiency on synthetic data and realistic videos for gesture and action classification.

1. Introduction

Spread of portable multimedia devices, e.g., smartphone, leads to enormous amount of videos produced everyday. For instance, in every minute, 100 h of videos are uploaded to YouTube [1]. Intelligent video classification system benefits video indexing, search, analysis, etc. Conventional pipeline of video classification involves in global video representations encoded by local visual descriptors. Despite of efficiency, such classification manner is applicable only to frame-level annotated video clips for both training and testing; in other words, the used videos should only contain one targeted action or event, from the start frame to the end, without unrelated video frames. However, frame-level annotation requires costly human intelligence and is too time-consuming, thereby, large amount of training videos are inaccessible. Besides, testing videos rely on delicate pre-processing schemes that divide them into clips (e.g., temporal clustering [2]), which prevents these approaches from practical applications. Even with well cut video clips, pooling step (e.g., average pooling [3] or max pooling [4]) in generating holistic representation would unavoidably bring in reductant and noisy information from background, which sometimes overwhelms representative visual patterns of targeted video class. In fact, local features in a small spatial-temporal window could be discriminative enough to indicate presence of one action or event in videos [5,6]. On the other hand, it is more convenient and easier to annotate an entire video rather than individual frames [7], especially the video corresponds to several semantic classes. In terms of web videos, people usually label videos using semantic tags, which means massive data with video-level annotation is available for training.

Inspired by the facts above, *multiple-instance learning* (MIL) based approach seems to be a straightforward option. In conventional MIL setting, training labels are given to sets of samples (*bags*) instead of individual samples (*instances*) [8]. The instance labels are given implicitly; a bag is positive if and only if at least one of its instances are positive. In recent years, MIL has enjoyed increasing popularity in computer vision. For example, in image classification and retrieval [9–11], each image is considered a bag. Different image regions are the instances. Instead of annotating each individual region, one only needs to label each training image with contained object names. MIL has also been applied to image segmentation [12,13] and tracking [14].

* Corresponding author. Fax: +1 732 4450537.
E-mail address: jl1322@cs.rutgers.edu (J. Liu).

http://dx.doi.org/10.1016/j.cviu.2015.10.012
1077-3142/

Fig. 1. Top row: hand rolling. Bottom row (hand crossing): the 3rd frame has similar visual patterns as hand rolling.

Specifically for video classification, each video sequence is considered as a bag and its segments as instances. As aforementioned, instead of labeling every frame, human experts only need to give the class types of each video, which allows people training video classifiers under MIL setting. However, most previous MIL-based methods assume the instances in a bag are independent and identically distributed (i.i.d.). Such assumption is problematic in many computer vision problems, particularly for tasks involved in sequential data. For video classification, consecutive frames or segments within a video tend to contain similar actions or events. Temporal correlations sometime are the key indicators of an action or an event. As shown in Fig. 1, if we consider each instance as independent, the 3rd frame in the bottom row will be mislabeled as hand rolling (the top row). This mistake can be avoided if we incorporate the temporal domain structure. In MIL literature, these structural information have not been fully explored. Some previous works [15–17] build edges between instances (within a same bag or cross bags) with similar appearance. These edges do enforce label consistency between similar instances; nevertheless they are not as powerful as edges in structured prediction models, e.g., the Conditional Random Fields (CRFs) [18]. CRFs exploit structures among data by modeling conditional distribution of observations and predictions, rather than joint distribution as used in Hidden Markov Model (HMM) [19]. In particular for video classification, the model could learn statistic dependencies existing in rich visual features and could identify an entire activity or event occurring in video clips; or to say, it is capable of make consistent predictions for consecutive frames. Furthermore, modeling label correlations of the intrinsic structure is more natural than enforcing all instances with similar appearance to have similar labels, especially we have explicitly know videos correspond to chain structure in temporal domain.

In this paper, we propose Weakly Supervised Sequence Modeling (WSSM) for video classification, a novel weakly supervised algorithm that combines MIL and chain CRFs in a seamless fashion. Our model takes each entire video as a bag and one video segment as an instance. It works in the MIL weakly supervised setting while inheriting the power of CRFs in modeling video sequences. The discriminative visual patterns for the targeted video class are explored through MIL fashion, while intrinsic temporal dependencies between video segments are explicitly modeled by chain CRFs. To estimate the CRFs' parameters, we formulate a new conditional likelihood which is only based on the video-level labels of training data. This conditional likelihood can be efficiently maximized using an alternating optimization method with a guaranteed convergence. The advantages of this algorithm are twofold. First, the CRF framework enables our model to outperform existing MIL methods in both bag and instance level label predictions for sequential data. Second, it is as easy as the traditional CRF framework in terms of both implementation and efficiency. Our algorithm has no extra parameters except for the regularizer weight, as all other CRF models have. Experiments show that training time of the proposed method is almost linear to the training data size.

Our approach is different from previous ones in both theory and application scenario. Deselaers and Ferrari [20] used CRF to solve classical MIL problems. Their method is not designed for structured data, thus cannot be used to incorporate temporal information for video classification. Zha et al. modeled instance labels as hidden variables of the CRFs [21]. However, solving the problem requires marginalizing over the whole space of hidden label configurations, whose size is exponential to the number of instances. Therefore, to solve the problem one has to use Gibbs sampler and convergence divergence at testing and training stages, making the solution inefficient and inaccurate. Several papers have borrowed MIL idea for video classification, for instance, in action recognition [6,22], event recognition [23], and sign language understanding [24,25]. However, temporal dependencies between video segments (instance) are not investigated in these methods.

2. Related work

2.1. Video classification

Video classification includes several key techniques. The first related technique is feature extraction, i.e., mining salient spatio-temporal features from videos that is highly involved in actions or events. One approach that achieves success in video classification is extracting HOG and HOF descriptors [26] from Space-Time Interest Points (STIP) [27] or dense trajectories [28,29]. Another important technique refers to feature encoding that generates video representations for based on local visual descriptors. Conventionally, after orthogonal feature transforming, e.g., Principle Component Analysis (PCA) [30], a codebook is constructed to quantize and aggregate feature vectors, where K-means and GMM [31] are commonly used. Then some encoding scheme is applied to the quantized vector to obtain high dimensional features. For example, boosting based feature ensembles [32], Fisher vector [33], and Vector of Locally Aggregated Descriptors (VLAD) [34] have illustrated more effectiveness. Very recently, Convolutional Neural Network (CNN) is also studied to implement feature extraction and encoding under deep neural network, and shows promising results on video classification [35,36]. Additionally, many efforts on feature pooling, normalization, and fusion have been also made on video classification.

Generally speaking, methods mentioned above mostly focus on devising global video representations, which only allow training and testing under fully supervised setting. Some others believed that local spatio-temporal visual is strong enough to differentiate videos. They learned classifiers with weakly supervised annotations, i.e., label is only given to the entire video. Latent Support Vector Machine (Latent SVM) has been applied to localize regions in the video to captures discriminative essence of the action class [37]. Some pioneering work that applied MIL is on action recognition [6,22], event recognition [23], and sign language understanding [24,25].

2.2. Multiple-instance learning

There a large body of work on multiple-instance learning. MIL was first proposed by Dietterich et al. [8] for drug activity prediction. A pioneering work is the *diversity density* (DD) by Maron and Lozano-Pérez [38]. This method used the co-occurrence of similar instances from different positive bags to learn the appearance of a positive instance. Zhang and Goldman [39] extended DD to an Expectation Maximization based method (EM-DD). Chen et al. [40] proposed MILES which maps bags into an instance-based feature space. The instance basis is selected using DD. Fu et al. [41] followed the basic idea of MILES, but intertwined the instance selection and classifier learning in an iterative manner, which improves the effectiveness of MILES. Andrews et al. [42] proposed two MIL algorithms, i.e., mi-SVM and MI-SVM, based on support vector machines (SVMs). Both of mi-SVM and MI-SVM learn instance-level classifiers. The first method, mi-SVM, maximizes the instance-level margins. The second one, MI-SVM, selects a *witness* instance from each bag and uses them to maximize the bag-level margins. By introducing a prior distribution of instance labels, Gehler and Chapelle [43] presented ALP-SVM which improves the empirical loss of the mi-SVM algorithm. The optimization problem is solved using deterministic annealing (DA). Other SVM-based MIL approaches include MI-kernel [44] and MICA [45]. Besides SVM, other learning models have been adapted for the MIL setting. Examples include k-nearest neighbor (kNN) [46], boosting [47], semi-supervised learning (SSL) [48], multi-label learning (MLL) [11] and random forest (RF) [49]. Most existing MIL algorithms use iterative or heuristic strategy in the training stage. Several works [50,51] relaxed the formulations into convex ones, so that the problem can be solved efficiently.

In recent years, the assumption that instances are generated i.i.d. has been dropped and the structures between instances have been exploited. Zhou et al. [15] constructed a graph for each bag. Instances with similar appearance were connected with edges. Graph kernels were used to train a classifier of bags. Babenko et al. [16] modeled each bag as a low dimensional manifold embedded in the feature space. Instances are samples from the manifold and can be used to reconstruct the manifold. The geometry of these manifolds is used for bag classification. Both works used structures to characterize bags. They cannot be used to predict instance labels. Vezhnevets et al. [52] used the MIL setting in the semantic segmentation task. Instances, i.e., superpixels, from different bags (images) are connected if they share similar appearance. Together with the original adjacency graph of the superpixels, a large graph containing superpixels of all training images are constructed and used for label prediction. Their method is well designed for the image segmentation task, and thus is difficult to be extended to other structured data. Zhang et al. [17] used a Laplacian-inspired graph regularizer to enforce labels of adjacent nodes/instances to be consistent. The problem is solved using concave–convex constraint programming and cutting plane techniques. Both of these two MIL methods can only enforce adjacent nodes to take similar labels. This is much less powerful in modeling structural information compared to CRFs.

3. Our method

We propose a new weakly supervised method for video classification, namely Weakly Supervised Sequence Modeling (WSSM). Our method is based on the customary CRFs formulation, yet training is under MIL setting rather than customary fully supervised fashion. In Section 3.1, we give detailed formulation of our model. At the training stage (Section 3.2), we formulate a new conditional likelihood function which only requires bag labels for entire videos. We use an alternating optimization method to optimize the likelihood and estimate the parameters. Our training algorithm is very efficient and is guaranteed to converge.

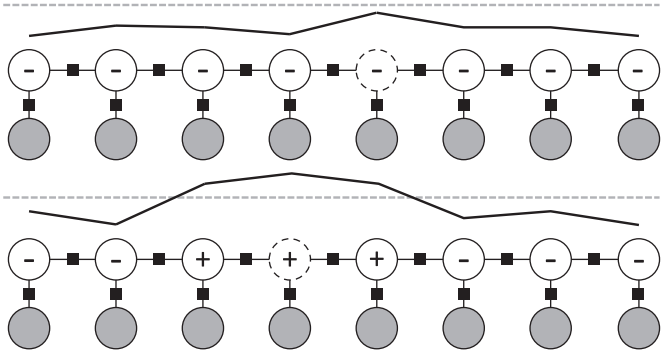

Fig. 2. Graphical model of chain CRFs: negative sequence (top) and positive sequence (bottom). Gray circles are observations x and white circles denote labelings y. Black squares indicate factors, including edge factors and node factors, in the mathematic form of $\exp(\langle w, \phi(x,y)\rangle)$. $\phi(x,y)$ is conventionally regarded as feature function. The two curves correspond marginal probability of each node being labeled '+'. Dotted line represents the probability threshold that is used to classify nodes into negative or positive. The dotted white circles are witness nodes used in training of proposed WSSM. In both positive and negative sequences, node with largest marginal probability is chosen as the witness node.

3.1. Formulation and inference

Our model follows a standard setting of Conditional Random Fields. We denote each video clip by a graph $G = (\mathcal{V}, \mathcal{E})$, as shown in Fig. 2. Each node/instance $i \in \mathcal{V}$ are consecutive video segments taking either positive or negative labels, i.e., related to a video class or not. Denote by $L = \{+1, -1\}$ the label set. We construct a conditional distribution over the space of all label configurations, $\mathcal{Y} = L^{\text{card}(\mathcal{V})}$. We call each label configuration a *labeling*. Given a parameter vector w and the observation of a video x, the conditional probability of a labeling y is:

$$p(y|x,w) = \frac{1}{Z(x,w)} \exp(\langle w, \phi(x,y)\rangle) \text{ in which}$$
$$Z(x,w) = \sum_{y\in\mathcal{Y}} \exp(\langle w, \phi(x,y)\rangle) \quad (1)$$

where $Z(x,w)$ is the *partition function* and $\phi(x,y)$ is the *feature function*. One may marginalize over all labelings to compute the *marginal* probability of a single node i being labeled ℓ, formally,

$$p_i(\ell|x,w) = \sum_{y\in\mathcal{Y}: y_i=\ell} p(y|x,w). \quad (2)$$

Similarly we have the edge marginal of an edge $(i,i') \in \mathcal{E}$:

$$p_{(i,i')}((\ell,\ell')|x,w) = \textstyle\sum_{y\in\mathcal{Y}: y_i=\ell, y_{i'}=\ell'} p(y|x,w). \quad (3)$$

Given a conditional random field, one predicts the labeling of instances by computing the *maximum a posteriori (MAP)*, $y^* = \text{argmax}_{y\in\mathcal{Y}}\, p(y|x,w)$. Alternatively, one can compute the *maximum marginal prediction* which assigns to each node the label with the largest marginal, $y^* = (\text{argmax}_{\ell\in L}\, p_i(\ell|x,w))_{i\in\mathcal{V}}$. In this paper, we choose the latter for our instance label predictor. At the bag level, we say a bag is positive if and only if at least one of its instances is labeled positive. Using marginals to predict nodes' labels leads to a convenient way to approximate the conditional probability of bag labels. This is the key to the new conditional likelihood we will propose in the training stage (Section 3.2). Since videos are sequential data with underlying chain structure, chain CRFs are explored. In computation, marginals and the maximum marginal prediction can be computed exactly and efficiently using dynamic programming (DP) when the underlying graph G is a chain.

3.2. Training

Denote by $\mathcal{D} = \{(x^n, y^n)\}_{n=1}^N$ the training data containing N bags (video clips). Each bag has an observation x^n and a bag label $y^n \in L$. Denote by $q(y^n|x^n, w)$ the conditional probability of bag n being labeled y^n. Assuming all bags are independently generated, using Bayes' rule, we have the conditional likelihood:

$$p(w|\mathcal{D}) = p(w) \prod_{n=1}^{N} \frac{q(y^n|x^n, w)}{q(y^n|x^n)} \tag{4}$$

where we use a Gaussian prior $p(w)$. Taking the negative log of the likelihood and dropping the term that is independent of w, we have the regularized loss function:

$$\begin{aligned}\mathcal{L}(w) &= \lambda\|w\|^2 - \sum_{n=1}^{N} \log q(Y^n|x^n, w) \\ &= \lambda\|w\|^2 - \sum_{n\in I^+} \log q(+1|x^n, w) - \sum_{n\in I^-} \log q(-1|x^n, w)\end{aligned} \tag{5}$$

where λ is the weight of the regularizer, I^+ and I^- are the index sets of positive bags and negative bags, respectively. We estimate the parameter vector by minimizing the loss function:

$w^* = \text{argmin}_w \mathcal{L}(w)$.

Since we use maximum marginal prediction, the marginal of each instance decides its label. We call an instance the *witness* if it has the maximal positive marginal within the bag. Intuitively, the witness is the "most positive" instance of the bag. By definition, a bag is labeled positive if and only if its witness is labeled positive. Therefore, the probability of the bag being positive can be approximated by the positive marginal of the witness $q(+1|x^n, w) = \max_{i\in\mathcal{V}^n} p_i(+1|x^n, w)$, in which $\mathcal{V}^n$ is the node set of bag n. Substituting the bag probability into the regularized log function, since $q(-1|x^n, w) = 1 - q(+1|x^n, w)$, we have:

$$\begin{aligned}\mathcal{L}(w) = \lambda\|w\|^2 &- \sum_{n\in I^+} \log \max_{i\in\mathcal{V}^n} p_i(+1|x^n, w) \\ &- \sum_{n\in I^-} \log\left(1 - \max_{i\in\mathcal{V}^n} p_i(+1|x^n, w)\right)\end{aligned} \tag{6}$$

We can use an alternating optimization method to minimize the loss function. We alternate the following two steps: (1) for fixed w, for each bag, find the witness instance, $\widehat{i}_n = \text{argmax}_{i\in\mathcal{V}^n} p_i(+1|x^n, w)$; (2) for fixed witness instances, minimize the loss function:

$$\begin{aligned}\widehat{\mathcal{L}}(w) = \lambda\|w\|^2 &- \sum_{n\in I^+} \log p_{\widehat{i}_n}(+1|x^n, w) \\ &- \sum_{n\in I^-} \log\left(1 - p_{\widehat{i}_n}(+1|x^n, w)\right)\end{aligned} \tag{7}$$

However, notice that at step (1) of every iteration, choosing the new witness for each bag will increase the corresponding node marginal $p_{\widehat{i}_n}(+1|x^n, w)$. In the loss function, the contribution of positive bags decreases. But the contribution of negative bags increases. Consequently the optimization is very inefficient. In order to gain computation advantages, we modify the loss function by making the contributions from negative bags independent of the witness indices. Note that a bag is negative if and only if all its instances are negative. For each negative bag, we replace the probability of the bag being negative, $q(-1|x^n, w)$, with the joint conditional probability of the labeling y^n being all -1's, $p(\vec{-1}|x^n, w)$. After the modification, the loss function to minimize at step (2) becomes:

$$\widetilde{\mathcal{L}}(w) = \lambda\|w\|^2 - \sum_{n\in I^+} \log p_{\widehat{i}_n}(+1|x^n, w) - \sum_{n\in I^-} \log p(\vec{-1}|x^n, w) \tag{8}$$

To find an initial w, we train w using a fully supervised chain CRFs, in which we assign negative labels to all instances in a negative bag and assign positive labels to all instances in a positive bag. See Algorithm 1 for pseudocode of the algorithm.

This algorithm is very efficient in practice as we will show in Section 4. We can also prove it converges.

Theorem 1. *Algorithm 1 converges.*

Algorithm 1: WSSM Training

Input: Training videos $\mathcal{D} = \{(x^n, y^n)\}_{n=1}^N$

Initialization: Train a fully supervised chain CRFs to initialize the parameter vector as $w^{(0)}$. Assign negative (positive) labels to all instances within a negative (positive) bag.

Repeat: in iteration τ

- **Step 1:** For $n \in I^+$, use $w^{(\tau)}$ to compute the marginal of each instance and update witness instance $\widehat{i}_n$.
- **Step 2:** $w^{(\tau+1)} = \text{argmin}_w \widetilde{\mathcal{L}}(w)$ (as defined in Eq.(8)).

Until The loss function converges.

Output: The parameter vector $w \in \mathbb{R}^M$ (M is the dimension of both edge and node feature).

Proof. At Step (2), we optimize the loss using gradient descent. Therefore, the loss monotonically decreases. It suffice to prove that at step (1), the loss also monotonically decreases. Denote by $\widehat{i}_n^*$ as the witness from the previous iteration. We select $\widehat{i}_n$ as the instance with the one with the maximal positive marginal. Therefore the positive marginal of $\widehat{i}_n$ is no smaller than that of $\widehat{i}_n^*$. The contribution of the bag n to the loss does not increase. □

Optimization details. We conclude this section by explaining how to implement the optimization Steps (1) and (2). Step (1) can be achieved by computing marginals exactly, since the graph is a tree devised for videos. Step (2) can be achieved using gradient descent. In practice, we use a Quasi-Newton optimization technique, e.g. L-BFGS [53]. It remains to show how to compute the gradient of the loss (Eq. (8)). We compute the gradient as follows (see Appendix A for more details):

$$\begin{aligned}\nabla_w \widetilde{\mathcal{L}}(w) = 2\lambda w &+ \sum_{n=1}^{N} \mathbb{E}_{y^n\sim p(y^n|x^n,w)}\left[\phi(x^n, y^n)\right] \\ &- \sum_{n\in I^+} \mathbb{E}_{y^n\sim p\left(y^n|x^n,w,y^n_{\widehat{i}_n}=+1\right)}\left[\phi(x^n, y^n)\right] - \sum_{n\in I^-1} \phi(x^n, \vec{-1})\end{aligned} \tag{9}$$

In the first summation, each summand is the expectation of the feature vector $\phi(x^n, y^n)$, where y^n follows the conditional distribution $p(y^n|x^n, w)$. In the second summation, each summand is also the expectation of the feature vector $\phi(x^n, y^n)$. But y^n follows the conditional distribution $p(y^n|x^n, w, y^n_{\widehat{i}_n} = +1)$, namely, the conditional distribution $p(y^n|x^n, w)$ under an extra condition that y^n has value $+1$ at the witness instance $\widehat{i}_n$. Due to the Markov property of CRFs, the first expectation can be factorized as follows:

$$\begin{aligned}\mathbb{E}_{y^n\sim p(y^n|x^n,w)}\left[\phi(x^n, y^n)\right] &= \sum_{i\in\mathcal{V}^n}\sum_{\ell\in L} p_i(\ell|x^n, w)\phi_i(x^n, y^n_i) \\ &+ \sum_{(i,i')\in\mathcal{E}^n}\sum_{\ell\in L}\sum_{\ell'\in L} p_{(i,i')}((\ell, \ell')|x^n, w)\phi_{(i,i')}(x^n, y^n_{(i,i')})\end{aligned} \tag{10}$$

where $p_i(\ell|x^n, w)$ and $p_{(i,i')}((\ell, \ell')|x^n, w)$ are node and edge marginals, respectively. We denote by $\phi_i(x^n, y^n_i)$ and $\phi_{(i,i')}(x^n, y^n_{(i,i')})$ node and edge features, respectively.

All marginals can be computed efficiently since the graph is a tree (chain). Similarly we can factorize and compute the second expectation $\mathbb{E}_{y^n\sim p(y^n|x^n,w,y^n_{\widehat{i}_n}=+1)}[\phi(x^n, y^n)]$, except that the marginals are computed by marginalizing over all labelings y^n with $+1$ at the witness instance $\widehat{i}_n$. This can be achieved by running DP with the hard constraint that the witness is positive.

4. Experiments

We evaluated the proposed WSSM algorithm on both synthetic sequential data and two real world video datasets. The implementation

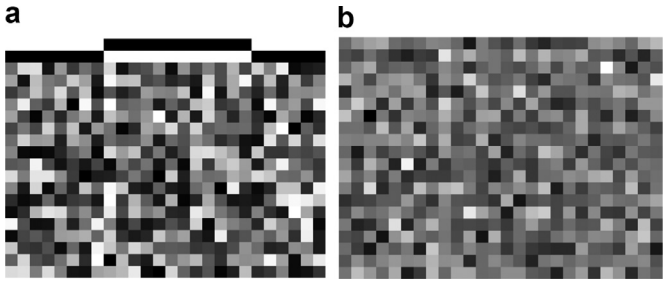

Fig. 3. An example synthetic data (positive): (a) original feature matrix; (b) corrupted feature matrix. Each column corresponds to the feature of one instance.

of WSSM was based on the UGM package[1]. We compared WSSM with several state-of-the-art MIL methods, such as mi-SVM [42], MI-SVM [42], MILES [40], mi-Graph [15], AL-SVM [43] and AW-SVM [43]. All of the experiments were conducted on a DELL PC with an eight-core 3.4 GHz CPU and 16 GB memory.

4.1. Synthetic experiments

We first evaluated WSSM on synthetic sequential data. Each sequence (bag) was generated as a chain of length K, which was randomly chosen between 20 and 40. We used the standard binary label setting: each instance is labeled either negative or positive. We first generated ground truth labels for instances, and the bag label was then determined by definition. Ground truth labeling was created as follows: creating a $2 \times K$ matrix of uniformly random values and then smoothing it along rows by convolving it with a Gaussian filter. This is to ensure the labeling of a sequence have correlations along the chain. The label of each instance was then given by the larger element in each column. Feature vectors for each node were created as the indicator vector of its label. We extended the two-dimensional feature vectors to 20 dimensions by attaching additional random noise values. See Fig. 3(a) for an example of the generated feature. In order to corrupt the feature with noise, we added i.i.d. Gaussian noise of standard deviation σ_s to the feature matrix (Fig. 3(b)).

We evaluated our method on such synthetic data of different noise levels ($\sigma_s = 0.1, 0.3, 0.5, 0.7, 1.0$). We introduced another parameter ρ, which measures the ratio of the number of positive instances over the size of the bag. Experimental results with different ρ could demonstrate how our method behaves when the amount of positive instances is small, medium and large within each positive bag. We run our experiments for the case when all bags have ρ within the ranges [20%, 30%], [45%, 55%] and [70%, 80%], respectively.

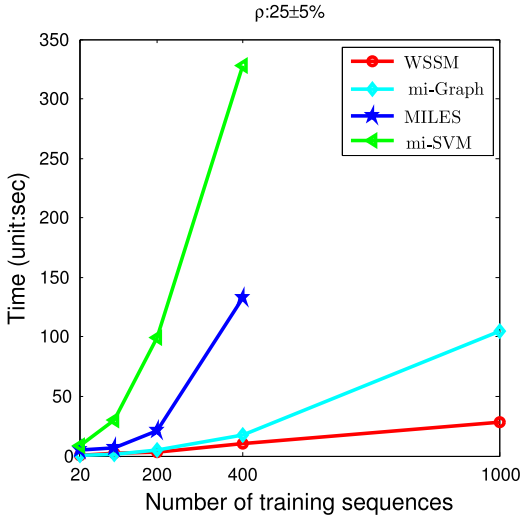

Fig. 5. Training time of different methods with $\sigma_s = 0.5$. For 1000 training samples, MILES runs out of the PC's memory; time for mi-SVM $\gg$ 1000 s.

For each given noise level and ρ, we randomly generated 50 positive and 50 negative sequences for training. For testing we produced 100 positive and 100 negative sequences. Comparisons of bag prediction accuracies are shown in Fig. 4. Obviously, WSSM outperforms other methods, especially with large noise level, e.g. $\sigma_s = 1$. The advantage is more obvious when ρ is small, i.e., WSSM performs relatively better when the ratio of positive instances within positive bags is small. We also compared WSSM with a baseline method denoted as WSSM$*$, which discards edge features in the original WSSM framework. The gap between the performance of WSSM$*$ and WSSM shows how much the structural information helps.

Running time analysis. In Theorem 1, we have proven that the loss of WSSM monotonously decreases and the optimization procedure is guaranteed to converges. We validated this argument in our experiments and further discuss the scalability of the WSSM model. In the aforementioned experiments, given 100 synthetic training data, WSSM converges within 4–8 iterations. We further compared the training time of WSSM with that of mi-Graph, MILES, and

[1] http://www.di.ens.fr/~mschmidt/Software/UGM.html

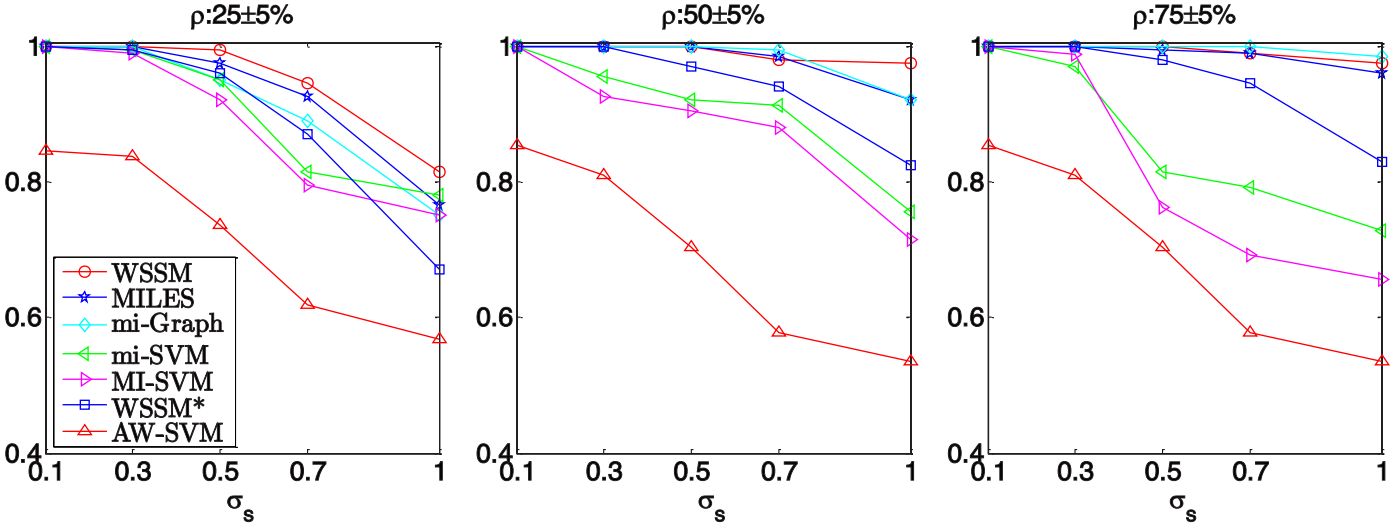

Fig. 4. Comparison of bag classification with $\sigma_s = 0.1, 0.3, 0.5, 0.7, 1$ and $\rho = 25 \pm 5\%, 50 \pm 5\%, 75 \pm 5\%$. Y-axis is the classification accuracy. We report the average over 10 runs.

Fig. 6. Gesture examples in ChaLearn gesture dataset. The top row are RGB images and the bottom shows the corresponding depth images (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.).

Table 1
The classification accuracy on both video-level and frame-level on the gesture recognition dataset. Some methods cannot be used for instance-level label prediction.

Method	WSSM (%)	Initialized WSSM (%)	mi-Graph (%)	MILES (%)	mi-SVM (%)	MI-SVM (%)	AL-SVM (%)	AW-SVM (%)
Video-level	**81.7**	66.8	76.7	76.1	68.1	56.4	64.6	75.1
Frame-level	**87.4**	41.9	–	–	81.2	84.8	–	–

mi-SVM, using different sizes of training data (20, 100, 200, 400 and 1000). The result is shown in Fig. 5. With growing training data size, the training time of mi-Graph increases quadratically, since it constructs graph kernels based on instance similarities. MILES uses diversity density for the selection of instance prototypes, which is costly in both time and memory. It runs out the PC memory with 1000 training data. The hyperplane learning in mi-SVM is also time-consuming. With 1000 training data, mi-SVM could not convergence within half an hour. WSSM is significantly more efficient than all other methods: WSSM takes only 28.7 s when training data size is 1000. The running time seems to be linear with regard to the training data size. WSSM is clearly more ready to be applied to dataset of large size.

4.2. Gesture recognition data

We evaluated the WSSM algorithm on the ChaLearn gesture recognition dataset 2011 [54]. This dataset consists of 20 video batches (devel 01–20), each of which includes 47 RGB video sequences and 47 corresponding depth videos recorded with the Kinect camera. In each video sequence, one actor made 1–5 gestures drawn from 8 to 15 gesture vocabularies (Fig. 6 shows several gesture examples in RGB image space and depth channel). The local features employed were HOG and HOF descriptors [28] from both RGB and depth images, based on STIP detector [27]. Finally, each video segment of 30 frame-length was represented by a 60-dimensional bag-of-word (BOW) feature vector. One-vs.-others classification was conducted for each batch independently. Within each batch, the first 30 video sequences was used for training and the rest for testing. We ignored the gestures with too few positive training bags (less than 5 video sequences) to avoid unbalance data issue.

The comparison of classification accuracies on video-level with more than 900 testing samples are shown in Table 1. Since the inference of WSSM is exactly the same as CRF model, we investigated the frame-level accuracy as well. Labels of video segments were assigned according to whether the segment contains frames related to targeted gestures. The results of initialized WSSM came from the parameter vector $w^{(0)}$ in Algorithm 1. It was learned using a fully supervised CRF model, with all instances in positive bags labeled as positive. Comparing WSSM with initialized WSSM, we could observe the amount of improvement attributed to our new conditional likelihood formulation. Notice that mi-Graph gets the second place that also explores the instance correlations through feature level similarities. However, its classification accuracy on video level is still 5% lower than WSSM.

4.3. Action recognition data

We further evaluate our method on the UCF-101 action recognition dataset [55], which contains real action videos of 101 classes collected from YouTube. Snapshots for the 101 action classes are shown in Fig. 7. This dataset is more challenging than ChaLearn gesture dataset, due to the more complex video contents and large intra-class variation. We first extracted improved Dense Trajectory (iDT) features [29] from each frame and each video segment of 15 frame-length was represented by 3000-dimension BOW feature vector (we catenated bow-of-features of HOG, HOF and MBH into one vector). We reported the classification results on train/test split 1 of UCF 101 that includes 13,320 videos. 9,537 videos were used in training and the other 3,783 for testing.

As shown in Fig. 5, the time complexity of most state-of-art MIL methods are higher than $O(n^2)$ (n is the training data size). In addition, visual features used to represent video segments are high dimensional. Thereby, these methods cannot train models in affordable computational time. On this dataset, we only compared our method with mi-Graph that has been regarded as state-of-art MIL method and shown good performance for structured data. We conducted one vs. others classification as well. In more detail, 101 CRF models were trained for the 101 action classes. For each class, all its training videos were used as positive bags, while we randomly chose 5 videos from each of other 100 classes to produce 500 negative bags.

The classification performance is evaluated by accuracy cross all the 101 classes. In term of overall accuracy, WSSM achieves 62.5%, compared to 57.4% of mi-Graph. Among the 101 action classes, WSSM performs better than mi-Graph on 58 ones, while being worse than its opponent on 34 classes. These results illustrate that with training data of weakly annotated labeling, WSSM achieves advantageous results than mi-Graph on video classification, by explicitly considering sequence structure. Fig. 8 shows the corresponding classification accuracy per-class.

Our method did not perform as well as state-of-art approaches reported on UCF 101, such as [56,57]. These are several reasons. First of all, the videos in UCF 101 are manually trimmed, such that each video corresponds to only one action class and every frame in the

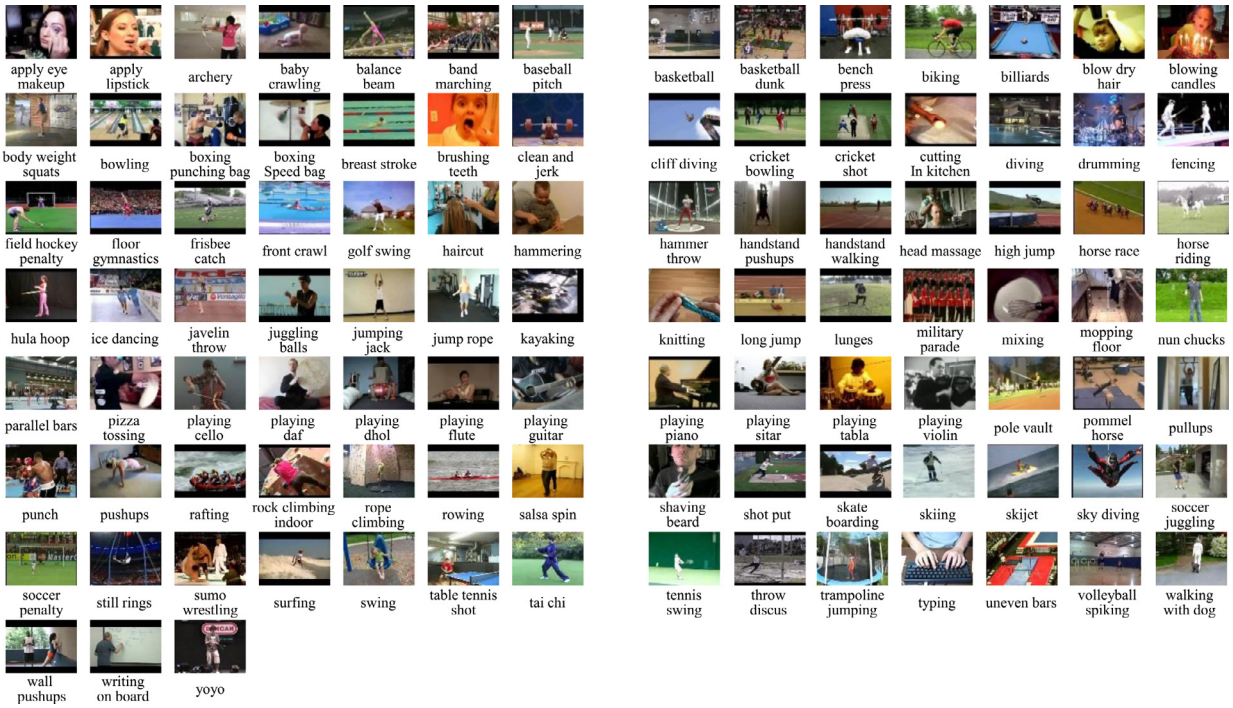

Fig. 7. Snapshots of action videos from UCF 101. Each one corresponds to one action class.

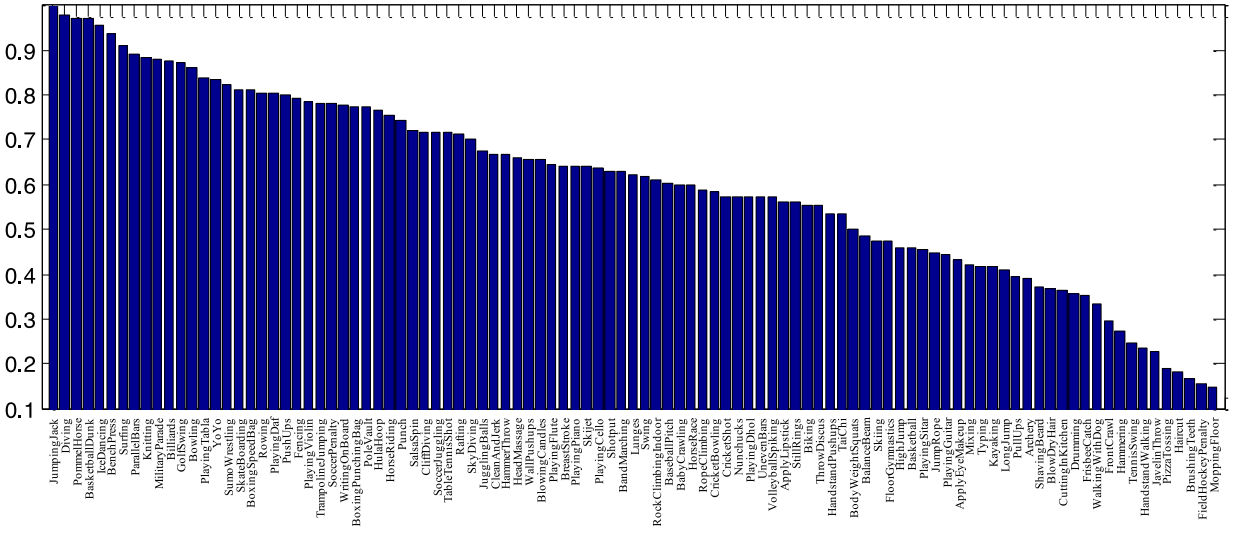

Fig. 8. Classification accuracy per-class on split 1 of UCF 101.

video involves in that action. This is why these methods could aggregate visual features from an entire video, without bringing in too much noise information. In other words, UCF 101 is tailored for training models with fully supervised labeling rather than weakly supervised setting we are targeting at. Secondly, we did not use all videos for training, considering data balance between positive and negative sequences. We speculate if we extend current WSSM to the multi-class fashion that can leverage all training videos, the performance could be improved. The last but not least, compared to these methods, the proposed WSSM algorithm is more applicable to the following scenarios: (1) it can leverage large amount of untrimmed videos for training; (2) training videos used to learn the model could contain multiple action classes.

5. Conclusions

In this paper, we proposed a new multiple-instance learning algorithm (WSSM) that seamlessly incorporates CRF model into the weakly supervised setting. Previous MIL methods explored structural information by enforcing label consistency of similar instance. Our WSSM models the conditional distribution of all label combinations of adjacent instances, which is more natural and more powerful. To estimate the CRF parameters, we designed a new conditional likelihood, customized for the setting when only bag labels are given for training. Node marginals of CRFs are used to formulate the conditional likelihood. Such likelihood can be efficiently maximized using an alternating optimization method with a guaranteed convergence.

We demonstrated the power of WSSM on both synthetic data and real video classification tasks on gesture and action recognition. Experiments show that our method outperforms existing MIL methods in video classification and also show promising results on frame-level prediction. Moreover, it has good merits in terms of both implementation and efficiency. WSSM has no extra parameters except for the regularizer parameter, and its training time in practice is almost linear to the training data size.

In this paper, we mainly focused on chain CRFs for video classification. However, our WSSM algorithm is readily applicable to data of more sophisticated structures, e.g. trees and loopy graphs, by the benefits of marginals. We would like to extend out model to multiple-label CRF. We would also like to investigate the possibility of using multiple modes [58,59] to compute marginals, in order to get a more robust result.

Acknowledgments

This research is partially supported by the Grants NSF IIS 1451292 and NSF CNS 1229628.

Appendix A

$$\widetilde{\mathcal{L}}(w) = \lambda\|w\|^2 - \sum_{n\in I^-}\log\frac{\exp\{\langle w,\phi(x^n,-\vec{1})\rangle\}}{\sum_{\bar{y}^n\in\mathcal{Y}}\exp\{\langle w,\phi(x^n,\bar{y}^n)\rangle\}} - \sum_{n\in I^+}\log\frac{\sum_{y^n\in\mathcal{Y}:y^n_{i_n}=+1}\exp\{\langle w,\phi(x^n,y^n)\rangle\}}{\sum_{\bar{y}^n\in\mathcal{Y}}\exp\{\langle w,\phi(x^n,\bar{y}^n)\rangle\}}$$

$$\begin{aligned}\nabla_w\widetilde{\mathcal{L}}(w) &= 2\lambda w + \sum_{n=1}^{N}\frac{\sum_{y^n\in\mathcal{Y}}\exp\{\langle w,\phi(x^n,y^n)\rangle\}\phi(x^n,y^n)}{\sum_{\bar{y}^n\in\mathcal{Y}}\exp\{\langle w,\phi(x^n,\bar{y}^n)\rangle\}} - \sum_{n\in I^-}\phi(x^n,-\vec{1}) \\ &\quad - \sum_{n\in I^+}\frac{\sum_{y^n\in\mathcal{Y}:y^n_{i_n}=+1}\exp\{\langle w,\phi(x^n,y^n)\rangle\}\phi(x^n,y^n)}{\sum_{\bar{y}^n\in\mathcal{Y}:\bar{y}^n_{i_n}=+1}\exp\{\langle w,\phi(x^n,\bar{y}^n)\rangle\}} \\ &= 2\lambda w + \sum_{n=1}^{N}\sum_{y^n\in\mathcal{Y}}p(y^n|x^n,w)\phi(x^n,y^n) - \sum_{n\in I^-}\phi(x^n,-\vec{1}) \\ &\quad - \sum_{n\in I^+}\sum_{y^n\in\mathcal{Y}}p\Big(y^n|x^n,w,y^n_{i_n}=+1\Big)\phi(x^n,y^n)\end{aligned}$$

References

[1] YouTube statistics. https://www.youtube.com/yt/press/statistics.html.
[2] F. Zhou, F. De la Torre, J.K. Hodgins, Hierarchical aligned cluster analysis for temporal clustering of human motion, Trans. Pattern Anal. Mach. Intell. 35 (3) (2013) 582–596.
[3] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, 2006, pp. 2169–2178.
[4] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 1794–1801.
[5] T.H. Thi, L. Cheng, J. Zhang, L. Wang, S. Satoh, Structured learning of local features for human action classification and localization, Image Vis. Comput. 30 (1) (2012) 1–14.
[6] M. Sapienza, F. Cuzzolin, P. Torr, Learning discriminative space–time action parts from weakly labelled videos, Int. J. Comput. Vis. 110 (1) (2014) 30–47.
[7] S. Bhattacharya, F.X. Yu, S.-F. Chang, Minimally needed evidence for complex event recognition in unconstrained videos, in: Proceedings of the International Conference on Multimedia Retrieval ICMR, 2014, p. 105.
[8] T.G. Dietterich, R.H. Lathrop, T. Lozano-Pérez, Solving the multiple instance problem with axis-parallel rectangles, Artif. Intell. 89 (1997) 31–71.
[9] C. Yang, T. Lozano-Pérez, Image database retrieval with multiple-instance learning techniques, in: Proceedings of Sixteenth International Conference on Data Engineering (ICDE), 2000, pp. 233–243.
[10] Y. Chen, J.Z. Wang, Image categorization by learning and reasoning with regions, J. Mach. Learn. Res. 5 (2004) 913–939.
[11] Z. Zhou, M. Zhang, Multi-instance multi-label learning with application to scene classification, in: Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems (NIPS), 2006, pp. 1609–1616.
[12] A. Vezhnevets, J.M. Buhmann, Towards weakly supervised semantic segmentation by means of multiple instance and multitask learning, in: Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 3249–3256.
[13] Y. Xu, J. Zhu, E. Chang, Z. Tu, Multiple clustered instance learning for histopathology cancer image classification, segmentation and clustering, in: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 964–971.
[14] B. Babenko, M.-H. Yang, S.J. Belongie, Visual tracking with online multiple instance learning, in: Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 983–990.
[15] Z. Zhou, Y. Sun, Y. Li, Multi-instance learning by treating instances as non-I.I.D. samples, in: Proceedings of the Twenty-Sixth Annual International Conference on Machine Learning (ICML), 2009.
[16] B. Babenko, N. Verma, P. Dollár, S. Belongie, Multiple instance learning with manifold bags, in: Proceedings of the Twenty-Fourth International Conference on Machine learning (ICML), 2011, pp. 81–88.
[17] D. Zhang, Y. Liu, L. Si, J. Zhang, R.D. Lawrence, Multiple instance learning on structured data, in: Proceedings of the Annual Conference on Advances in Neural Information Processing Systems (NIPS), 2011.
[18] J.D. Lafferty, A. McCallum, F.C.N. Pereira, Conditional random fields: probabilistic models for segmenting and labeling sequence data, in: Proceedings of the Eighteenth International Conference on Machine Learning (ICML), 2001, pp. 282–289.
[19] L.R. Rabiner, B.-H. Juang, An introduction to hidden Markov models, IEEE Acoust. Speech Signal Process. Mag. 3 (1) (1986) 4–16.
[20] T. Deselaers, V. Ferrari, A conditional random field for multiple-instance learning, in: Proceedings of the Twenty-Seventh International Conference on Machine Learning (ICML), 2010, pp. 287–294.
[21] Z. Zha, X. Hua, T. Mei, J. Wang, G. Qi, Z. Wang, Joint multi-label multi-instance learning for image classification, in: Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition CVPR, 2008.
[22] S. Ali, M. Shah, Human action recognition in videos using kinematic features and multiple instance learning, Trans. Pattern Anal. Mach. Intell. 32 (2) (2010) 288–303.
[23] K.-T. Lai, F. Yu, M.-S. Chen, S.-F. Chang, Video event detection by inferring temporal instance labels, in: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition CVPR, 2014, pp. 2251–2258.
[24] T. Pfister, J. Charles, A. Zisserman, Large-scale learning of sign language by watching TV (using co-occurrences), in: Proceedings of the 2013 British Machine Vision Conference BMVC, 2013.
[25] N. Michael, P. Yang, Q. Liu, D.N. Metaxas, C. Neidle, A framework for the recognition of nonmanual markers in segmented sequences of American sign language., in: Proceedings of the Twenty-Second British Machine Vision Conference BMVC, 2011, pp. 1–12.
[26] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, in: Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition CVPR, 2008, pp. 1–8.
[27] I. Laptev, On space-time interest points, Int. J. Comput. Vis. 64 (2-3) (2005) 107–123.
[28] H. Wang, A. Kläser, C. Schmid, C.-L. Liu, Dense trajectories and motion boundary descriptors for action recognition, Int. J. Comput. Vis. 103 (1) (2013) 60–79.
[29] H. Wang, C. Schmid, Action recognition with improved trajectories, in: Proceedings of the 2013 IEEE International Conference on Computer Vision ICCV, 2013, pp. 3551–3558.
[30] I. Jolliffe, Principal Component Analysis, Wiley Online Library, 2005.
[31] C.M. Bishop, et al., Pattern Recognition and Machine Learning, vol. 1, Springer, New York, 2006.
[32] A. Oikonomopoulos, I. Patras, M. Pantic, Spatiotemporal localization and categorization of human actions in unsegmented image sequences, Trans. Image Process. 20 (4) (2011) 1126–1140.
[33] C. Sun, R. Nevatia, Large-scale web video event classification by use of Fisher vectors, in: Proceedings of the 2013 IEEE Workshop on Applications of Computer Vision WACV, 2013, pp. 15–22.
[34] H. Jégou, M. Douze, C. Schmid, P. Pérez, Aggregating local descriptors into a compact image representation, in: Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR, 2010, pp. 3304–3311.
[35] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, in: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition CVPR, 2014.
[36] Z. Wu, Y.-G. Jiang, J. Wang, J. Pu, X. Xue, Exploring inter-feature and inter-class relationships with deep neural networks for video classification, in: Proceedings of the ACM International Conference on Multimedia MM, 2014, pp. 167–176.
[37] N. Shapovalova, A. Vahdat, K. Cannons, T. Lan, G. Mori, Similarity constrained latent support vector machine: an application to weakly supervised action classification, in: Proceedings of the Twelfth European Conference on Computer Vision ECCV, Springer, 2012, pp. 55–68.
[38] O. Maron, T. Lozano-Pérez, A framework for multiple-instance learning, in: Proceedings of the Conference on Advances in Neural Information Processing Systems NIPS, 1998, pp. 570–576.
[39] Q. Zhang, S.A. Goldman, EM-DD: an improved multiple-instance learning technique, in: Proceedings of the 2001 Advances in Neural Information Processing Systems NIPS, 2001, pp. 1073–1080.

[40] Y. Chen, J. Bi, J. Wang, MILES: multiple-instance learning via embedded instance selection, Trans. Pattern Anal. Mach. Intell. 28 (12) (2006) 1931–1947.
[41] Z. Fu, A. Robles-Kelly, J. Zhou, MILIS: multiple instance learning with instance selection, Trans. Pattern Anal. Mach. Intell. 33 (5) (2011) 958–977.
[42] S. Andrews, I. Tsochantaridis, T. Hofmann, Support vector machines for multiple-instance learning, in: Proceedings of the Conference on Advances in Neural Information Processing Systems NIPS, 2002, pp. 561–568.
[43] P.V. Gehler, O. Chapelle, Deterministic annealing for multiple-instance learning, in: Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics AISTATS, 2007, pp. 123–130.
[44] T. Grtner, P.A. Flach, A. Kowalczyk, A.J. Smola, Multi-instance kernels, in: Proceedings of the Nineteenth International Conference on Machine Learning ICML, 2002, pp. 179–186.
[45] O.L. Mangasarian, E.W. Wild, Multiple Instance Classification via Successive Linear Programming, Technical Report 05-02, Data Mining Institute, 2005.
[46] J. Wang, J.-D. Zucker, Solving the multiple-instance problem: a lazy learning approach, in: Proceedings of the Seventeenth International Conference on Machine Learning ICML, 2000, pp. 1119–1126.
[47] P.A. Viola, J.C. Platt, C. Zhang, Multiple instance boosting for object detection, in: Proceedings of the Conference on Advances in Neural Information Processing Systems NIPS, 2005.
[48] R. Rahmani, S.A. Goldman, MISSL: multiple-instance semi-supervised learning, in: Proceedings of the Twenty-Third International Conference on Machine Learning ICML, 2006, pp. 705–712.
[49] C. Leistner, A. Saffari, H. Bischof, MIForests: multiple-instance learning with randomized trees, in: Proceedings of the Eleventh European Conference on Computer Vision ECCV, 2010, pp. 29–42.
[50] F. Li, C. Sminchisescu, Convex multiple-instance learning by estimating likelihood ratio, in: Proceedings of the Conference on Advances in Neural Information Processing Systems NIPS, 2010, pp. 1360–1368.
[51] Y.-F. Li, I.W. Tsang, J.T. Kwok, Z.-H. Zhou, Convex and scalable weakly labeled SVMs., J. Mach. Learn. Res. 14 (2013) 2151–2188.
[52] A. Vezhnevets, V. Ferrari, J.M. Buhmann, Weakly supervised semantic segmentation with a multi-image model, in: Proceedings of the 2011 International Conference on Computer Vision ICCV, 2011, pp. 643–650.
[53] D.C. Liu, J. Nocedal, On the limited memory BFGS method for large scale optimization, Math. Program. 45 (1-3) (1989) 503–528.
[54] ChaLearn gesture challenge. https://sites.google.com/a/chalearn.org/gesturechallenge/, 2011.
[55] K. Soomro, A.R. Zamir, M. Shah, Ucf101: A Dataset of 101 Human Action Classes From Videos in the Wild, 2012. arXiv preprint arXiv:1212.0402
[56] X. Peng, L. Wang, X. Wang, Y. Qiao, Bag of visual words and fusion methods for action recognition: comprehensive study and good practice, arXiv:1405.4506(2014).
[57] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: Proceedings of the Twenty-Seventh Conference on Advances in Neural Information Processing Systems NIPS, 2014, pp. 568–576.
[58] C. Chen, V. Kolmogorov, Y. Zhu, D. Metaxas, C.H. Lampert, Computing the M most probable mode of a graphical model, in: Proceedings of the International Conference on Artificial Intelligence and Statistics AISTATS, 2013.
[59] C. Chen, H. Liu, D. Metaxas, T. Zhao, Mode estimation for high dimensional discrete tree graphical models, in: Proceedings of the Twenty-Seventh Conference on Advances in Neural Information Processing Systems NIPS, 2014, pp. 1323–1331.