Sum-Product Networks for Modeling Activities with Stochastic Structure

Mohamed R. Amer and Sinisa Todorovic
Oregon State University
amerm@onid.orst.edu, sinisa@eecs.oregonstate.edu

Abstract

This paper addresses recognition of human activities with stochastic structure, characterized by variable space-time arrangements of primitive actions, and conducted by a variable number of actors. We demonstrate that modeling aggregate counts of visual words is surprisingly expressive enough for such a challenging recognition task. An activity is represented by a sum-product network (SPN). SPN is a mixture of bags-of-words (BoWs) with exponentially many mixture components, where subcomponents are reused by larger ones. SPN consists of terminal nodes representing BoWs, and product and sum nodes organized in a number of layers. The products are aimed at encoding particular configurations of primitive actions, and the sums serve to capture their alternative configurations. The connectivity of SPN and parameters of BoW distributions are learned under weak supervision using the EM algorithm. SPN inference amounts to parsing the SPN graph, which yields the most probable explanation (MPE) of the video in terms of activity detection and localization. SPN inference has linear complexity in the number of nodes, under fairly general conditions, enabling fast and scalable recognition. A new Volleyball dataset is compiled and annotated for evaluation. Our classification accuracy and localization precision and recall are superior to those of the state-of-the-art on the benchmark and our Volleyball datasets.

1. Introduction

Suppose that a video shows "unloading of a car," and that we want to recognize this class of person-car interactions. Recognition in this case is challenging, because unloading can be done in many distinct ways. For example, while unloading the trunk, the person might answer the cell phone, or fetch stuff from the back seat. Now, suppose the video shows a volleyball rally (e.g., setting the ball to the left). Recognition challenges are even more pronounced in this example because of many alternative sequences of ball passes between the players defining the set type.

In this paper, we address recognition of activities with multiple, alternative spatiotemporal structures, like in the examples above. Their variability primarily arises from the different numbers of actors and objects they interact with, and the different temporal arrangements of primitive actions conducted by the actor(s). This problem has received scant attention in the literature. We also present a new Volleyball dataset, which abounds with such activities, and thus is suitable for our evaluation and future benchmarking.

It is widely acknowledged that explicit modeling of activity structure helps recognition [25, 1, 23]. Graphical models have been successfully used for this purpose, including HMMs and Dynamic Bayesian Networks [3, 27], prototype trees [13, 7], AND-OR graphs [6, 21], and Markov Logic Networks [22, 2]. On datasets with structure-rich activity classes (e.g., UT interactions [19]), graphical models typically have superior performance over representations which do not explicitly capture activity structure, such as Bag-of-Words (BoW) [11], and probabilistic Latent Semantic Analysis (pLSA) and Latent Dirichlet Allocation (LDA) [15]. Recognition rates may increase even further by grounding graphical models onto object-detector responses [10], rather than using raw video features (e.g., HOGs).

Although we address activities with distinctive spatiotemporal structures, we depart from the above prior work. Our key intuition is that, locally, our activities do not have informative structure. This is because of the variable numbers of actors, and a wide range of ways primitive actions can be conducted. Therefore, the "structureless" BoWs seem suitable for modeling particular video parts. However, a configuration of primitive actions comprising an activity will give rise to spatiotemporal changes of BoWs. For example, a change in histograms of visual words will occur when "walking" is followed by "jumping." Consequently, in addition to modeling histograms of codewords of certain video parts, it is also necessary to model their spatiotemporal changes across the video. We show that this can be done efficiently by capturing constraints of codeword histograms across different video parts, characterizing the activity.

A video is represented by a regular, space-time grid of points, as illustrated in Fig. 1a. Every point on the grid is characterized by a distribution of counts of visual words oc-

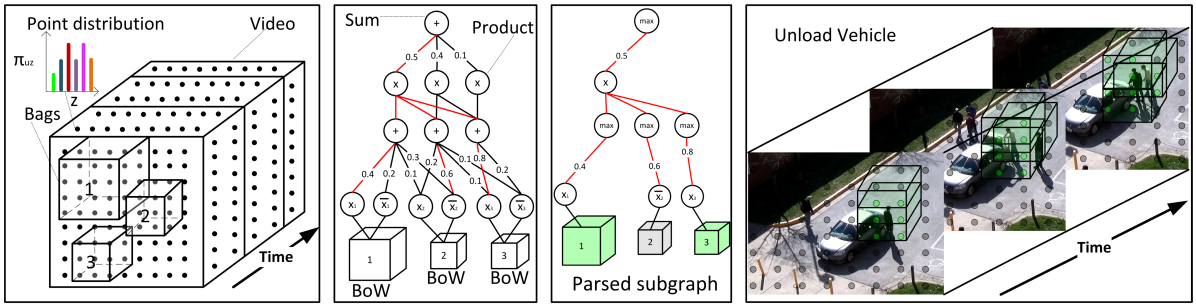


Figure 1. Our approach: (a) A video is represented by the counting grid of visual words; every grid point is assigned a distribution of word counts; BoWs are placed across the counting grid and characterized by aggregate distributions of word counts on the grid. (b) Our activity model is the sum-product network (SPN) that consists of layers of sum and product nodes, ending with BoWs at terminal nodes; children nodes in the SPN can be shared by multiple parents. (c) SPN inference amounts to parsing, and identifying foreground BoWs (green boxes). (d) Localization of the activity "unloading of the trunk" in an example sequence from the VIRAT dataset [16].

curring at that point. We refer to this grid model as *counting grid* [17, 8]. A large set of averaging windows are placed across the counting grid (Fig. 1a). The windows serve to estimate distributions of codeword counts over the corresponding video parts. Foreground windows are selected such that they jointly capture spatiotemporal changes of codeword distributions characterizing the activity (Fig. 1c).

Each averaging window is treated as a disordered BoW. The selected set of foreground BoWs mutually constrain the counts of visual words that fall within each bag. These joint constraints on word counts arise from the spatiotemporal structure of the activity. Surprisingly, and as a fundamental contribution of this paper, our results demonstrate that modeling the joint constraints of codeword counts across the video is expressive enough to capture the stochastic structure of rather complex activities. This is surprising, as it contrasts an established school of thought that, for activities with distinct structure, explicit modeling of space-time relations between video features is indispensable.

To model alternative configurations of BoWs, we use the sum-product network (SPN), illustrated in Fig. 1b. SPN is a generalized directed acyclic graph, consisting of three types of nodes – namely, terminal, sum, and product nodes [18]. In our approach, the terminal nodes represent BoWs placed at different locations in the video. The sums correspond to mixtures over different subsets of the terminals, and the product nodes correspond to mixture components.

SPN is suitable for capturing alternative structures of an activity, as desired. This is because the product nodes can encode particular configurations of BoWs, whereas the sum nodes can account for alternative configurations. In addition, SPN can compactly encode an exponentially large number of alternative arrangements of BoWs. This is because SPN consists of a number of hidden layers of sums and products, where children nodes are shared by parents. Thus, SPN can be viewed as a compact way to specify a mixture model with exponentially many mixture components, where subcomponents are reused by larger ones. We represent each activity of interest by one SPN model.

When a new video is encountered, we place a large set of BoWs across the video's counting grid. Then, we establish the BoWs as terminal nodes of each SPN representing one of the target activities. SPN inference amounts to parsing the SPN graph, i.e., selecting a subset of optimal sum, product, and terminal nodes (i.e., BoWs) from the graph that jointly yield the highest most probable explanation (MPE) of activity occurrence (Fig. 1c). The video is assigned the label of the activity whose SPN yields the maximum MPE. The selected subset of BoWs localize foreground video parts that are relevant for recognition (Fig. 1d).

Unlike other "deep" models (e.g., AND-OR graphs [21], and convolutional networks [12]), SPN inference is *exact*, and has linear complexity in the number of nodes, under fairly general conditions [18], which are not restrictive for our purposes. Consequently, SPNs enable fast, and scalable activity recognition.

In the following, Sec. 2 explains our contributions; Sec. 3 formulates the SPN; Sections 4 and 5 specify our inference and learning algorithms; Sec. 6 describes our feature extraction; and Sec. 7 presents our empirical evaluation.

2. Our Contributions Relative to Prior Work

Our key contributions include: (i) new activity representation—SPN; (ii) extension of the counting grid (CG) model [17, 8] to a hierarchical mixture of BoWs; and (iii) new Volleyball dataset.

To our knowledge, SPNs have never been used for activity recognition. In [18], SPN is specified for binary random variables at the terminal nodes. We here generalize SPN by grounding its terminal nodes onto histograms of visual words associated with BoWs. SPNs are related to AND-OR graphs, which are also capable of encoding alternative

configurations of subactivities within an activity [6, 21]. An AND-OR graph is defined as a normalized product of potential functions. Consequently, inference and learning of AND-OR graphs is usually intractable, due to the normalizing partition function which is a sum of an exponential number of products. By contrast, SPN enables exact computation of the partition function in linear time in the number of SPN nodes. SPNs are also related to deep belief networks (DBNs) [12]. DBNs are constructed by stacking many layers of graphical models, which reduces their learning to parameter estimation. By contrast, learning SPNs involves estimating both the graph connectivity, and parameters of the model. While DBNs are not probabilistic, SPNs can be viewed as probabilistic, general-purpose convolution networks, with max-pooling corresponding to Most Probable Explanation (MPE) inference.

The CG is aimed at capturing constraints of feature histograms across different parts of a still image [17], or text documents [8], where the constraints arise from the fact that the image features (or document features) live in an ordered 2D space (or multidimensional space). The CG is formulated as a *product* of a few independent BoWs. By contrast, we use a more expressive mixture model of BoWs, with exponentially many mixture components of a large number of BoWs. Also, our model additionally selects an optimal subset of foreground BoWs from a large set of candidates for activity localization.

Video representations aggregating local features over space-time regions have been explored by several methods; however, they focused only on single-actor actions [9, 26], punctual and repetitive actions [24], or activities with deterministic sequence of primitive actions [14].

3. The Model

This section, first, explains how to ground SPN onto BoWs, and then formulates SPN.

3.1. The Counting Grid and BoWs

Given a dictionary, $\mathcal{Z}$=$\{$z$\}$, of visual words, we assume that the distribution of word occurrences in the video is governed by the SPN mixture of BoWs. In particular, every point u on a regular grid spanning the video (see Fig. 1a) probabilistically generates one visual word from the distribution $\boldsymbol{\pi}_u = [\pi_{uz}]$. Each π_{uz} represents the probability of occurrence of word z at point u on the grid, such that $\forall u$, $\sum_z \pi_{uz} = 1$.[1] The point distributions $\boldsymbol{\pi}_u$ are used to probabilistically generate a set of BoWs, $\mathcal{H} = \{H_b : b = 1, 2, ..., n\}$, with space-time volumes $|H_b|$, and at different locations across the video. A bag H_b can be generated in two steps — first, by placing an averaging window over the corresponding set of $|H_b|$ points on the grid; and, second, by probabilistically sampling a word at every point u that falls within the window, using the distribution of word occurrences within the bag $\frac{1}{|H_b|}\sum_{u\in H_b}\pi_{uz}, \forall z \in \mathcal{Z}$.

We formalize this generative process by the following likelihood of counts of visual words generated by bag H_b:

$$P(\boldsymbol{c}_b|H_b) \propto \prod_z \big[\sum_{u\in H_b}\pi_{uz}\big]^{c_{bz}}, \tag{1}$$

where $\boldsymbol{c}_b = [c_{bz}]$ are counts of visual words observed in H_b, and $\propto$ means proportional up to the normalizing constant.

Given a large set of bags $\mathcal{H}$ in the video, we expect that only a subset of BoWs will coincide with video parts that are relevant for activity recognition, i.e., foreground. Let X_b denote a binary random variable that H_b belongs to foreground, characterized by the Dirichlet prior $P(X_b{=}1) \propto \prod_z[\sum_{u\in H_b}\pi_{uz}]^{\theta_z-1}$ with parameters $\boldsymbol{\theta} = [\theta_z]$. Then, from (1), the posterior probability that H_b captures the activity-specific counts of visual words is defined as

$$P_{X_b|\boldsymbol{c}_b} = P(X_b{=}1|\boldsymbol{c}_b) \propto \prod_z \big[\sum_{u\in H_b}\pi_{uz}\big]^{c_{bz}+\theta_z-1}. \tag{2}$$

In the sequel, SPN is formulated as a hierarchical mixture of BoWs from $\mathcal{H}$.

3.2. Mixture of BoWs

The SPN is a rooted directed acyclic graph (see Fig. 1b), with terminal, sum, and product nodes. The terminal nodes $\boldsymbol{X} = \{(X_b, \overline{X_b}) : b = 1, ..., n\}$ are the binary indicators of corresponding BoWs. They are aimed at selecting foreground BoWs from $\mathcal{H}$ as components of the mixture, where $X_b = 1$ and its inverse indicator $\overline{X_b} = 0$ denote that H_b is selected, otherwise $X_b{=}0$ and $\overline{X_b}{=}1$. Instantiations of random variables $\boldsymbol{X}$ are denoted with small letters: $X_b = x_b \in \{0, 1\}$ and $\overline{X_b} = \overline{x_b} \in \{0, 1\}$.

Without loss of generality, we will assume that the sums and products are arranged in alternating layers, i.e., all children of a sum are products or terminals, and all children of a product are sums. Indices of the sum nodes are denoted with i, l, and indices of the product nodes are denoted with j, k. We use i^+ to denote the set of children of i.

An edge (i, j) that connects sum i with its child j has a non-negative weight $w_{ij} \geq 0$, where $\sum_{j\in i^+} w_{ij} = 1$. Also, edges (i, j) that connect product k with its children $l \in k^+$, have uniform weights $w_{kl} = 1$. This allows us to recursively define the values of sum and product nodes, given the observed counts of visual words in the video $\mathcal{C} = \{\boldsymbol{c}_b : b = 1, ..., n\}$, as

$$S_i(\mathcal{C}) = \sum_{j\in i^+} w_{ij} S_j(\mathcal{C}), \quad S_k(\mathcal{C}) = \prod_{l\in k^+} S_l(\mathcal{C}). \tag{3}$$

At the lowest layer, the values of sum nodes connecting to the terminal nodes $\boldsymbol{X}$ are defined as

$$S_i(\mathcal{C}) = \sum_{b\in i^+}[w_{ib1}x_b P_{X_b|\boldsymbol{c}_b} + w_{ib2}\overline{x_b}(1 - P_{X_b|\boldsymbol{c}_b})], \tag{4}$$

[1] Since distinct activity classes are characterized by different spatiotemporal arrangements of visual words, the distributions $\boldsymbol{\pi}_u(a)$ are learned separately for each activity class $a \in \mathcal{A}$. In this section, we drop the explicit reference to the activity class in notation, for simplicity.

where $P_{X_b|\boldsymbol{c}_b}$ is the posterior probability, given by (2). Note that $S_i(\mathcal{C})$ includes both terminals X_b and $\overline{X_b}$ in the mixture with weights w_{ib1} and w_{ib2}, where $\sum_{b\in i^+}(w_{ib1}+w_{ib2})=1$.

From (3) and (4), the SPN rooted at sum node i can be viewed as a mixture model, whose children are the mixture components, which, in turn, are products of mixture models. The value of the SPN is the value of its root, $S(\mathcal{C})$.[2]

Note that setting both $X_b = 1$ and $\overline{X_b} = 1$ at the same time amounts to marginalizing out H_b from $S(\mathcal{C})$. Consequently, we can efficiently compute the joint posterior distribution of all bags $\mathcal{H}$ in the video as:

$$P(\boldsymbol{X}|\mathcal{C}) = S(\mathcal{C})/S_{\boldsymbol{X}=\mathbf{1}}, \tag{5}$$

where the normalizing constant $S_{\boldsymbol{X}=\mathbf{1}}$ is the value of the SPN when all indicators are set to 1, $\{X_1{=}1, \overline{X_1}{=}1, ..., X_n{=}1, \overline{X_n}{=}1\}$.

The joint posterior distribution of BoWs, given by (5), defines our model of an activity class. The model parameters include $\Omega = \{\{w_{ij}\}, \{\pi_{uz}\}, \{\theta_z\}\}$. Note that our mixture model in (5) is significantly more expressive than the counting grid of [8], specified as a product of all BoWs: $P(\mathcal{C}|\mathcal{H}) \propto \prod_b P(\boldsymbol{c}_b|H_b)$. In the following two sections, we specify inference and learning of our model.

4. Inference

Given a video, we extract video features at points of the counting grid, and map the features to a dictionary of visual words. Then, we place a large set of bags $\mathcal{H}$ across the grid, and compute the counts of visual words within every bag $\mathcal{C} = \{\boldsymbol{c}_b\}$. These counts are taken as observables of SPN, $S(\mathcal{C}; a)$, for activity class $a \in \mathcal{A}$. The goal of inference is to parse each activity model from the set of SPNs, $\{S(\mathcal{C}; a) : a \in \mathcal{A}\}$, and thus select a subset of sum, product, and terminal nodes which yield the highest most probable explanation (MPE) $\{\hat{S}(\mathcal{C}; a) : a \in \mathcal{A}\}$. The video is assigned the label of the activity class whose parsed SPN gives the maximum MPE: $\hat{a} = \text{argmax}_{a\in\mathcal{A}}\, \hat{S}(\mathcal{C}; a)$.

Our inference consists of two passes through the SPN graph. For estimating MPE, the SPN graph is modified, such that the sum nodes are replaced by maximizations, and the products are kept intact (see Fig. 1c). In the bottom-up pass, a max node outputs the maximum weighted value among its children, and a product node outputs the product of its children's values. Thus, from (3), the upward pass of inference computes:

$$\hat{S}_i(\mathcal{C}) = \max_{j\in i^+} w_{ij}\hat{S}_j(\mathcal{C}), \quad \hat{S}_k(\mathcal{C}) = \prod_{l\in k^+} \hat{S}_l(\mathcal{C}), \tag{6}$$

where the terminal indicators are set as $X_b = 1$ and $\overline{X}_b = 0$, $b = 1, ..., n$, to account for all BoWs in the video.

The top-down pass performs parsing. It starts from the root, and then recursively selects: (i) Child $\hat{j}$ of a max node i, previously selected in the bottom-up pass, whose marginal MPE is maximum, $\hat{j} = \text{argmax}_{j\in i^+} w_{ij}\hat{S}_j(\mathcal{C})$; and (ii) All children of a previously selected product node. At the terminals of the parsed SPN, we set:

$$\begin{array}{l} X_b{=}1, \overline{X}_b{=}0 \text{ , if } w_{ib1}P_{X_b|\boldsymbol{c}_b} \geq w_{ib2}(1{-}P_{X_b|\boldsymbol{c}_b}); \\ X_b{=}0, \overline{X}_b{=}1 \text{ , if } w_{ib1}P_{X_b|\boldsymbol{c}_b} < w_{ib2}(1{-}P_{X_b|\boldsymbol{c}_b}); \end{array} \tag{7}$$

This selects the foreground BoWs[3]. Finally, the resulting subgraph is used to compute the MPE value of the root node, $\hat{S}(\mathcal{C})$. Our inference is summarized in Alg. 1.

Algorithm 1: Inference

Input: Observed counts of visual words $\mathcal{C}$ in the input video. SPNs for activities $a \in \mathcal{A}$.
Output: Activity label $\hat{a}$, and foreground BoWs.

for $a \in \mathcal{A}$ **do**
  Upward pass: compute $\hat{S}_i(\mathcal{C})$ and $\hat{S}_k(\mathcal{C})$ using (6);
  Downward pass:
  - at a max node i: select $\hat{j}{=}\,\text{argmax}_{j\in i^+} w_{ij}\hat{S}_j(\mathcal{C})$;
  - at a product node: select all of its children;
  - at a terminal node: select the BoW as in (7);
end
$\hat{a} = \text{argmax}_{a\in\mathcal{A}}\, \hat{S}(\mathcal{C}; a)$

5. Learning

Given a set of training videos $\mathcal{T} = \{t\}$ showing instances of an activity class, our goal is to learn parameters $\boldsymbol{w} = [w_{ij}]$, $\boldsymbol{\pi} = [\pi_{uz}]$, and $\boldsymbol{\theta} = [\theta_z]$ of the SPN model. We here assume that we are given the counts of visual words $\mathcal{C}^t = \{\boldsymbol{c}_b^t\}$ of bags $\mathcal{H} = \{H_b\}$ placed in the same space-time layout in each training video $t \in \mathcal{T}$.

For learning, we use the EM algorithm that alternates two steps in each iteration τ: (i) *E-step:* Estimation of the expected SPN connectivity $\boldsymbol{w}^{(\tau+1)}$, given the parameters of the counting grid, $\boldsymbol{\pi}^{(\tau)}$ and $\boldsymbol{\theta}^{(\tau)}$, obtained in the previous iteration τ; and (ii) *M-step:* Estimation of $\boldsymbol{\pi}^{(\tau+1)}$ and $\boldsymbol{\theta}^{(\tau+1)}$, given the estimate of the SPN graph connectivity $\boldsymbol{w}^{(\tau+1)}$. Below, we will drop the explicit reference to iterations.

In the E-step, we run our inference Alg. 1 on $\mathcal{T}$, which produces distinct parses of the SPN for each $t \in \mathcal{T}$. Then,

[2] For example, the value of SPN in Fig. 1b is $S(\mathcal{C}) = 0.5(0.4x_1P_1 + 0.2\overline{x_1}\,\overline{P_1} + 0.1x_2P_2 + 0.3\overline{x_2}\,\overline{P_2})(0.2x_2P_2 + 0.6\overline{x_2}\,\overline{P_2} + 0.1x_3P_3 + 0.1\overline{x_3}\,\overline{P_3})(0.8x_3P_3{+}0.2\overline{x_3}\,\overline{P_3}){+}0.4(0.2x_2P_2{+}0.6\overline{x_2}\,\overline{P_2}{+}0.1x_3P_3{+}0.1\overline{x_3}\,\overline{P_3})(0.8x_3P_3{+}0.2\overline{x_3}\,\overline{P_3}){+}0.1(0.8x_3P_3{+}0.2\overline{x_3}\,\overline{P_3})$, where $P_b = P_{X_b|\boldsymbol{c}_b}$ and $\overline{P_b} = 1 - P_{X_b|\boldsymbol{c}_b}$, given by (2).

[3] When SPNs are complete and consistent, the conditions introduced in [18], their exact inference and learning are tractable (polynomial time). SPN is complete iff all children of a sum node have access to the same subset of $\boldsymbol{X}$ as that sum node. SPN is consistent iff no random variable in $\boldsymbol{X}$ appears both negated and non-negated as two children of a product node. In our case, completeness and consistency of SPN comes automatically by construction. If bag H_b violates completeness or consistency then our inference automatically sets $X_b{=}1, \overline{X}_b{=}1$, i.e., sums out H_b from S.

for every child j of sum nodes i, we maintain a count of j's occurrences in the resulting SPN parses. The weights $\{w_{ij}\}$ are then estimated by normalizing these counts for every sum node i. Since Alg. 1 is based on MPE, note that we conduct the "hard" E-step, which has been shown to successfully address learning of deep models with tens of layers in other applications [18].

In the M-step, we use the estimates of $\{(x_b^t, \overline{x_b^t})\}$ and $\boldsymbol{w}$ to compute $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$. Recall that parsing of the SPN by Alg. 1 selects foreground BoWs in each $t \in \mathcal{T}$. This amounts to instantiating the indicator variables $X_b = x_b^t \in \{0,1\}$ and $\overline{X_b} = \overline{x_b^t} \in \{0,1\}$ for each video $t \in \mathcal{T}$. In the M-step, we maximize a lower variational bound, V, of the log-distribution of the training data, L,

$$V \leq L = \textstyle\sum_t \log \hat{S}(\mathcal{C}^t). \tag{8}$$

Below, we present our final results of the variational maximization, while the derivation steps are given in Appendix.

From (3), (4) and (8), the expression of L includes the following non-constant terms that depend on $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$:

$$L \propto \textstyle\sum_t \log \sum_b \left[w_{ib1} x_b^t P_{X_b|\boldsymbol{c}_b^t} + w_{ib2} \overline{x_b^t} (1 - P_{X_b|\boldsymbol{c}_b^t}) \right]. \tag{9}$$

Then, from (2) and (9), the lower variational bound of L is

$$\begin{aligned} V \;=\; & \sum_t \Big[\sum_b Q_b \log[(w_{ib1} x_b^t - w_{ib2} \overline{x_b^t})/Q_b] \\ & + \sum_b Q_b \sum_z (c_{bz}^t + \theta_z - 1) \log \big[\sum_{u \in H_b} \pi_{uz} \big] \Big], \end{aligned} \tag{10}$$

where $\{Q_b : b = 1, ..., n\}$ is the variational distribution over the latent selection of foreground BoWs in the video. Maximizing V with respect to $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$, subject to the constraints that for all points u on the counting grid $\sum_z \pi_{uz} = 1$, and $\sum_z \theta_z = 1$, gives the following update rules:

$$\begin{aligned} \pi_{uz} \;&\propto\; \pi_{uz} \sum_t \sum_b \frac{Q_b (c_{bz}^t + \theta_z - 1)}{\sum_{u \in H_b} \pi_{uz}}, \\ \theta_z \;&\propto\; \theta_z \sum_b Q_b \sum_{u \in H_b} \frac{\pi_{uz} \log[\sum_{u \in H_b} \pi_{uz}]}{\sum_{u \in H_b} \pi_{uz}}. \end{aligned} \tag{11}$$

where Q_b that maximizes V can be computed, using the standard variational derivation of the exact E-step, as

$$Q_b \propto \exp[\sum_{t,z} (w_{ib1} x_b^t - w_{ib2} \overline{x_b^t})(c_{bz}^t + \theta_z - 1) \log[\sum_{u \in H_b} \pi_{uz}]]. \tag{12}$$

Note the parameters on the right-hand-side of (11) and (12) are estimates of the previous EM iteration.

Initialization: The initial distributions $\{\pi_{uz}^{(0)}\}$ and $\{\theta_z^{(0)}\}$ are estimated as the average counts of visual word occurrences observed at every point u of the counting grids associated with training videos $t \in \mathcal{T}$, such that $\sum_z \pi_{uz}^{(0)} = 1$, and $\sum_z \theta_z^{(0)} = 1$. The initial height of SPN is set to 8 layers. In each non-terminal layer, the SPN width is set to the same number of 10 nodes. At the lowest layer of sum nodes, we connect each sum to distinct, overlapping groups of BoWs. The groups are defined by a temporal extent that the bags cover in the video. Each group occupies 20% of the video, where the temporal displacement between neighboring groups of BoWs is 10% of the video (i.e., neighboring groups overlap by 50% in time), as illustrated in Fig. 2a. For the upper layers, we establish edges between nodes in consecutive layers, such that every child sees 20% of parents in the layer above. While establishing edges, we ensure that the initial SPN meets the requirements of completeness and consistency, for tractability [18].

Our learning is summarized in Alg. 2. In our experiments, Alg. 2 converges in about $\tau_{\max} = 10 - 20$ iterations, depending on the activity class. After learning, the final SPN graph is obtained by pruning edges with zero weights, and recursively removing non-root parentless nodes. Note that such pruning does not violate completeness and consistency of the resulting SPN.

Algorithm 2: Learning

Input: Training videos of an activity class $\mathcal{T} = \{t\}$
Output: $\Omega = \{\boldsymbol{w}, \boldsymbol{\pi}, \boldsymbol{\theta}\}$

```
Initialize SPN, and estimate w^(0), π^(0), θ^(0)
for τ = 1 : τ_max do
    for t ∈ T do
        Inference on t using Alg. 1;
    end
    Renormalize w^(τ) using the parsing results on T;
    Compute π^(τ) and θ^(τ) as in (11)
end
```

6. Feature Extraction

We use the state-of-the-art approach to feature extraction based on the two-layered Stacked Convolutional ISA network (SCISA), presented in [12]. We closely follow their setup, since it was thoroughly evaluated and demonstrated as superior against prior work. Specifically, we represent a video by a space-time grid of $N \times N \times N$ points that are evenly distributed along each of the spatial and time axes, where $N = 64$ by default. Then, a 3D patch of size (16 pixels) $\times$ (16 pixels) $\times$ (10 frames) at each grid point is input to the SCISA network to produce a 500-dimensional local feature. The resulting local feature is then mapped to the closest visual word in a 300-dimensional dictionary. As in [12], we train the SCISA network on 200K video patches, and learn the dictionary of visual words on the resulting local features using the K-means, with K=300.

Next, we place 3D bags (i.e., BoWs) centered at every

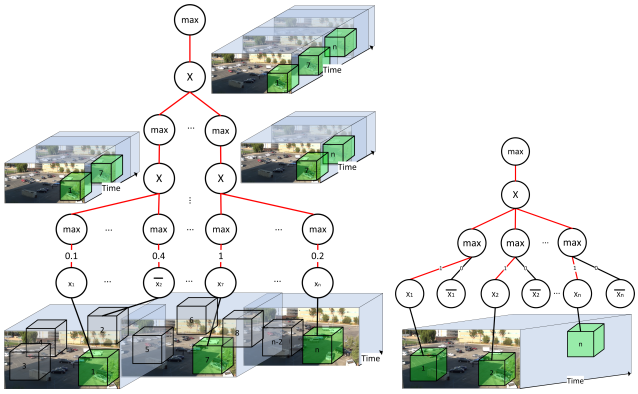
point of the counting grid. The bags enclose $\frac{N}{2^m} \times \frac{N}{2^m} \times \frac{N}{2^m}$ neighboring points (except bags on the video boundary), where $m = 2, 3, ..., \log_2 \frac{N}{4}$. We consider two strategies for specifying the bag size: (i) m is set to one specific value and applied to all bags; (ii) m is varied in the interval $[2, \log_2 \frac{N}{4}]$ producing a hierarchy of bags.

7. Results

We evaluate video classification accuracy, and recall and precision of localizing foreground video parts. A true positive is declared if the intersection of an identified foreground BoW and ground-truth bounding box is larger than 50% of their union. For evaluation, we use four datasets: VIRAT 1.0 & 2.0 [16], UT-Interactions [19], KTH [20], and our own new Volleyball dataset.

Datasets: *VIRAT 1.0 & 2.0* includes 11 activity classes: loading and unloading a vehicle, opening and closing the trunk of a vehicle, getting into and out of a vehicle, entering and exiting a facility, running, carrying an object, and gesturing. VIRAT is suitable for our evaluation, because its videos show activities with structural variations. As in [4], we have partitioned the VIRAT footage into shorter clips. Each clip shows an instance of one of the 11 activities, which may take between 5-10 seconds, as well as 10 additional seconds of the original longer video. These additional 10 seconds may occur randomly before and/or after the target activity instance. In this way, we have compiled a dataset with 20 VIRAT clips for each activity, where 50% are used for training, and 50%, for testing. *UT-Interactions* consists of 120 video clips, 20% for training and 80% for testing, where each clip shows an instance from the set of 6 activity classes: shaking-hands, pointing, hugging, pushing, kicking, and punching. *KTH* consists of 2391 short sequences showing 6 types of human actions: walking, jogging, running, boxing, hand waving, and hand clapping. Although our focus is on complex activities, we use KTH for comparison against the state of the art [12, 11, 19, 15] under their setup: videos of 16 subjects are used for training, and videos of other 9 subjects are used for testing. VIRAT and UT-Interactions provide ground-truth bounding boxes around all actors and objects participating in the activity.

Most existing benchmark datasets show relatively simple punctual and repetitive actions with little structural variability. Therefore, we have compiled a new *Volleyball dataset* for our evaluation. It consists of 240 videos showing 6 classes of volleyball set types – namely, setting the ball to the left, right, middle, back right, back left, and back middle (Fig. 3). Volleyball dataset is suitable for our purposes, since each activity class can be realized through a wide range of different sequences of ball passes between the volleyball players. There are 40 videos per class, randomly split into 20 training and 20 test videos. Each video is about 4 seconds long, and shows one instance of the volleyball set type at resolution 848×480. The videos show a large variability of players' movements under occlusion. We provide ground-truth annotations in terms of bounding boxes around the volleyball players engaged in the activity. The dataset and annotations are made public on our website.



(a) Inference of Var1 on a VIRAT sequence (b) Var3

Figure 2. An illustration of inference for Var1 and Var3. Var3 selects all BoW as foreground, and computes their product, which is equivalent to the counting grid model of [8].

Our Variants: To evaluate sensitivity to our design choices, we define a default variant of our approach, Var1, and then make changes in one computational step at a time, resulting in variants Var2 and Var3. **Var1** uses the counting grid with $N \times N \times N$ points, where $N = 64$. Every grid point is a center of a hierarchy of bags enclosing $(2^{-m}N) \times (2^{-m}N) \times (2^{-m}N)$ neighboring points, where $m = 1, 2, ..., \log_2 \frac{N}{4}$. At each grid point, Var1 extracts local features using the SCISA network of [12], and maps them to 300 visual words. The SPN height is 8, and the width is 10. **Var2** changes the feature extraction step, and uses the cuboid spatiotemporal features[5], which were also used in [15, 19]. A comparison of Var1 and Var2 tests our dependency on the type of local features used. **Var3** uses a simplified three-layered SPN, illustrated in Fig. 2. In Var3, all BoWs are selected as foreground, i.e., $\forall b$, $X_b = 1$ and $\overline{X_b} = 0$, and all BoWs are connected to their sum nodes at the upper layer with weights set to 1. There is only one product node connected to the root. A comparison of Var1 and Var3 tests the advantages of a deep mixture model versus the counting grid model.

Quantitative Results: Table 1 shows that our average classification accuracy is superior to that of Var2, Var3, and competing methods which use: (i) SVM of a single BoW of local features learned by SCISA [12]; (ii) SVM of space-time grids of local features [11]; (iii) SVM with a kernel that accounts for spatiotemporal matches of interest points [19]; and (iii) pLSA and LDA models [15]. Interestingly, even without deep learning of local features, Var2 outperforms the approaches of [12, 11, 19, 15]. Our Volleyball dataset is the most challenging due to small inter-class differences.

Dataset	Var1	Var2	Var3	[12]	[11]	[19]	[15]
KTH	95.2±2.5	94.1±2.3	91±4.2	93.9	91.8	91.1	83.3

Dataset	Var1	Var2	Var3	[12]	[3]	[19]
UT	82.4±2.6	80.3±2.3	75.2±5.3	76.0	75.7	70.8

Dataset	Var1	Var2	Var3	[12]
VIRAT	76.2±3.1	72.5±4	70.7±4.2	68.1
Volleyball	69.8±4.6	65.0±4.3	59.0±3.4	56.6

Table 1. Average classification accuracy in [%] on KTH, UT-Interactions, VIRAT, and Volleyball dataset.

	The SPN height					
Dataset	4	8	16	24	32	64
VIRAT	75.4±4.2	76.2±3.1	76.3±5.1	76.2±4.2	74.6±5.4	70.9±5.6
Volleyball	67.5±5.8	69.8±4.6	69.1±4.3	68±3.1	65±2.3	62.5±2.6

	The number of points in the counting grid is $N \times N \times N$			
Dataset	$N = 16$	$N = 32$	$N = 64$	$N = 128$
VIRAT	63.6±4.4	71.9±3.4	76.2±3.1	74.72±4.5
Volleyball	60.1±4.1	63.6±5.4	69.8±4.6	66.3±5.1

	The bag size is $(64 \cdot 2^{-m}) \times (64 \cdot 2^{-m}) \times (64 \cdot 2^{-m})$ grid points			
Dataset	$m = 2$	$m = 3$	$m = 4$	$m = 2, 3, 4$
VIRAT	68.1±3.4	72.72±5.3	74.5±3.7	76.2±3.1
Volleyball	52.5±4.8	61.1±4.4	63.3±5.1	69.8±4.6

Table 2. Average classification accuracy in [%] of Var1 using SPNs with different heights, different numbers of points in the counting grid, and different sizes of BoWs. Note that m=2, 3, 4 denotes that we use a hierarchy of BoWs.

	Bag size = $(2^{-m}64) \times (2^{-m}64) \times (2^{-m}64)$ grid points							
	$m = 2$		$m = 3$		$m = 4$		m=2, 3, 4	
Dataset	Var1	Var3	Var1	Var3	Var1	Var3	Var1	Var3
VIRAT	.97(.22)	.95(.20)	.90(.42)	.87(.39)	.73(.52)	.71(.42)	.70(.72)	.68(.58)
UT	.96(.18)	.92(.15)	.88(.31)	.84(.28)	.72(.51)	.65(.41)	.62(.70)	.60(.55)
Volleyball	.78(.26)	.72(.16)	.69(.29)	.64(.23)	.52(.33)	.46(.31)	.54(.52)	.48(.45)

Table 3. Recall and precision (in parentheses) for different sizes of BoWs.

Table 2 shows our sensitivity to specific choices of the number of: (i) layers in SPN, (ii) counting grid points, and (ii) grid points enclosed by each BoW. As can be seen, we are relatively insensitive to these parameter over a certain range of their values. As the SPN height and width increase, the results improve, at the price of increased computation time. After a certain model complexity, larger model heights and widths lead to overfitting. For Var1, we choose the smallest SPN height and width which give equally good performance as more complex models.

The recall and precision of our localization are given in Table 3. The highest F-measure is obtained when Var1 uses a hierarchy of bags with sizes defined by varying m=2, 3, 4.

Running time: Without feature extraction, inference by Var1 on the 4sec Volleyball videos takes less than 10s; and learning SPN on 20 training Volleyball videos takes about 700sec on a 2.66GHz, 3.49GB RAM PC.

Qualitative Results of our inference using Var1 are illustrated in Fig. 2a and Fig. 3. In particular, Fig. 2a shows a small excerpt of the parsed SPN model, and the inferred foreground BoWs. Fig. 3 illustrates our localization results on a few frames from the Volleyball and VIRAT videos.

8. Conclusion

We have addressed detection and localization of activities with stochastic structure. When activities have distinctive structure, prior work strongly argues the advantages of explicitly modeling spatiotemporal relations of activity parts. By contrast, we have shown that modeling a distribution of aggregate counts of visual words in the video is surprisingly expressive enough for our purposes. Our activity model is a sum-product network (SPN) of bags-of-words (BoWs). SPN represents a hierarchical mixture of distributions of counts of visual words, which are detected on a regular space-time grid in the video. SPN inference provides most probable explanation (MPE), and has linear complexity in the number of nodes, under the conditions of completeness and consistency, which can be ensured by construction. Consequently, SPNs enable fast, and scalable activity recognition and localization. The results demonstrate our superior performance over competing methods, and relative invariance to specific choices of the SPN height, width, number of points in the counting grid, and sizes of BoWs, over a certain range of their values. We have also compiled a new, challenging dataset of six types of volleyball rallies with variable spatiotemporal structures, and small inter-class differences.

Appendix

To derive the update rules of $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$, given by (11), we maximize the lower variational bound V of the log-distribution of data, given by (10). Note that the second term of (10) requires computing the summation $\sum_{u \in H_b} \pi_{uz}$ before applying the logarithm. We bound this second term by using the Jensen's inequality as

$$\log[\sum_{u \in H_b} \pi_{uz}] = \log[\sum_{u \in H_b} \frac{\pi_{uz} r_{uz}}{r_{uz}}] \geq \sum_{u \in H_b} r_{uz} \log[\frac{\pi_{uz}}{r_{uz}}] \tag{13}$$

where r_{uz} is a distribution over points u, $r_{uz} \geq 0$, $\sum_{u \in H_b} r_{uz} = 1$. r_{uz} can be computed as a function of π_{uz}, by constrained optimization of the objective in (13), resulting in $r_{uz} = \frac{\pi_{uz}}{\sum_{u \in H_b} \pi_{uz}}$. Consequently, from (10), we derive the variational bound

$$\begin{aligned} V \quad \geq \quad & \sum_t \Big[\sum_b Q_b \log[(w_{ib1} x_b^t - w_{ib2} \overline{x_b^t})/Q_b] \\ & + \sum_b Q_b \sum_z (c_{bz}^t + \theta_z - 1) \sum_{u \in H_b} r_{uz} \log(\frac{\pi_{uz}}{r_{uz}}) \Big]. \end{aligned} \tag{14}$$

Maximizing (14) with respect to π_{uz}, r_{uz}, and θ_z, subject to $\sum_z \pi_{uz} = 1$, $\sum_z \theta_z = 1$ and $\sum_{u \in H_b} r_{uz} = 1$, gives (11).

Acknowledgement

This research has been sponsored in part by NSF IIS 1018490, and DARPA MSEE FA 8650-11-1-7149.

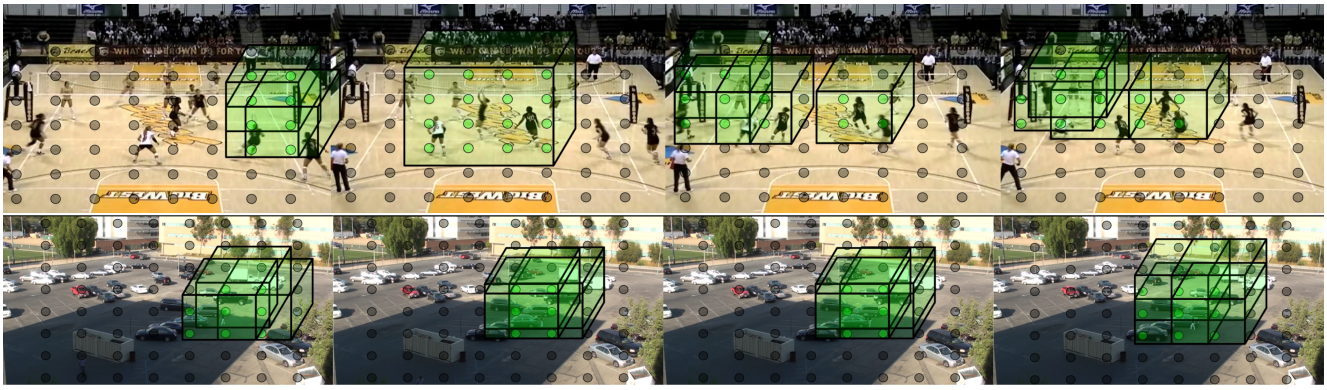


Figure 3. Localization of Var1: (top) An example video sequence from the Volleyball dataset showing the activity "setting the ball to the left." (bottom) An example video sequence from the VIRAT dataset showing the activity "loading of a vehicle." As can be seen, Var1 correctly estimates foreground BoWs (green bags), in both cases. We visualize only a subset of points of the counting grid, for clarity.

References

[1] J. Aggarwal and M. Ryoo. Human activity analysis: A review. *ACM Comput. Surv.*, 43:16:1–16:43, 2011.

[2] M. Albanese, R. Chellappa, N. Cuntoor, V. Moscato, A. Picariello, V. S. Subrahmanian, and O. Udrea. PADS: A probabilistic activity detection framework for video data. *IEEE TPAMI*, 32:2246–2261, 2010.

[3] M. Amer and S. Todorovic. A Chains model for localizing group activities in videos. In *ICCV*, 2011.

[4] S. Bhattacharya, R. Sukthankar, R. Jin, and M. Shah. A probabilistic representation for efficient large scale visual recognition tasks. In *CVPR*, 2011.

[5] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *IWVS-PETS*, 2005.

[6] A. Gupta, P. Srinivasan, J. Shi, and L. Davis. Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos. In *CVPR*, 2009.

[7] R. Hamid, S. Maddi, A. Bobick, and I. Essa. Structure from statistics: Unsupervised activity analysis using suffix trees. In *ICCV*, pages 1–8, 2007.

[8] N. Jojic and A. Perina. Multidimensional counting grids: Inferring word order from disordered bags of words. In *UAI*, 2011.

[9] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *CVPR*, 2010.

[10] T. Lan, Y. Wang, W. Yang, and G. Mori. Beyond actions: Discriminative models for contextual group activities. In *NIPS*, 2010.

[11] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[12] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR*, 2011.

[13] Z. Lin, Z. Jiang, and L. S. Davis. Recognizing actions by shape-motion prototype trees. In *ICCV*, 2009.

[14] J. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, 2010.

[15] J. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *IJCV*, 79(3):299–318, 2008.

[16] S. Oh and et al. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR*, 2011.

[17] A. Perina and N. Jojic. Image analysis by counting on a grid. In *CVPR*, 2011.

[18] H. Poon and P. Domingos. Sum-product networks: A new deep architecture. In *UAI*, 2011.

[19] M. S. Ryoo and J. K. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *ICCV*, 2009.

[20] C. Schueldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *ICPR*, 2004.

[21] Z. Si, M. Pei, B. Yao, and S.-C. Zhu. Unsupervised learning of event AND-OR grammar and semantics from video. In *ICCV*, 2011.

[22] S. D. Tran and L. S. Davis. Event modeling and recognition using Markov logic networks. In *ECCV*, 2008.

[23] P. K. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE Trans. Circuits Syst. Video Techn.*, 18(11):1473–1488, 2008.

[24] M. M. Ullah, S. N. Parizi, and I. Laptev. Improving bag-of-features action recognition with non-local cues. In *BMVC*, 2010.

[25] D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. In *CVIU*, volume 115, pages 224–241, 2011.

[26] X. Wu, D. Xu, L. Duan, and J. Luo. Action recognition using context and appearance distribution features. In *CVPR*, 2011.

[27] Z. Zeng and Q. Ji. Knowledge based activity recognition with Dynamic Bayesian Network. In *ECCV*, 2010.