

Unsupervised Learning and Segmentation of Complex Activities from Video

Fadime Sener, Angela Yao
 University of Bonn, Germany
 {sener,yao}@cs.uni-bonn.de

Abstract

This paper presents a new method for unsupervised segmentation of complex activities from video into multiple steps, or sub-activities, without any textual input. We propose an iterative discriminative-generative approach which alternates between discriminatively learning the appearance of sub-activities from the videos' visual features to sub-activity labels and generatively modelling the temporal structure of sub-activities using a Generalized Mallows Model. In addition, we introduce a model for background to account for frames unrelated to the actual activities. Our approach is validated on the challenging Breakfast Actions and Inria Instructional Videos datasets and outperforms both unsupervised and weakly-supervised state of the art.

1. Introduction

We address the problem of understanding complex activities from video sequences. A complex activity is a procedural task with multiple steps or sub-activities that follow some loose ordering. Complex activities can be found in instructional videos; YouTube hosts hundreds of thousands of such videos on activities as common as '*making coffee*' to the more obscure '*weaving banana fibre cloths*'. Similarly, in assistive robotics, a robot that can understand and parse the steps of a household task such as '*doing laundry*' can anticipate and support upcoming steps or sub-activities.

Complex activity understanding has received little attention in the computer vision community compared to the more popular simple action recognition task. In simple action recognition, short, trimmed clips are classified with single labels, *e.g.* of sports, playing musical instruments [10, 27], and so on. Performance on simple action recognition has seen a remarkable boost with the use of deep architectures [10, 25, 29]. Such methods however are rarely applicable for temporally localizing and/or classifying actions from longer, untrimmed video sequences, usually due to the lack of temporal consideration. Even works which do incorporate some modelling of temporal structure [4, 24, 28, 29] do little more than capturing frame-to-frame changes, which is why the state of the art still relies

on either optical flow [25] or dense trajectories [29, 30]. Moving towards understanding complex activities then becomes even more challenging, as it requires not only parsing long video sequences into semantically meaningful sub-activities, but also capturing the temporal relationships that occur between these sub-activities.

We aim to discover and segment the steps of a complex activity from collections of video in an unsupervised way based purely on visual inputs. Within the same activity class, it is likely that videos share common steps and follow a similar temporal ordering. To date, works in a similar vein of unsupervised learning all require inputs from narration; the sub-activities and sequence information are extracted either entirely from [17, 1], or rely heavily [23] on text. Such works assume that the text is well-aligned with the visual information of the video so that visual representations of the sub-activity are learned from within the text's temporal bounds. This is not always the case for instructional videos, as it is far more natural for the human narrator to first speak about what will be done, and then carry out the action. Finally, reliably parsing spoken natural language into scripts¹ is an unsolved and open research topic in itself. As such, it is in our interest to rely only on visual inputs.

In this work, we propose an iterative model which alternates between learning a discriminative representation of a video's visual features to sub-activities and a generative model of the sub-activities' temporal structure. By combining the sub-activity representations with the temporal model, we arrive at a segmentation of the video sequence, which is then used to update the visual representations (see Fig. 1a). We represent sub-activities by learning linear mappings from visual features to a low dimensional embedding space with a ranking loss. The mappings are optimized such that visual features from the same sub-activity are pushed together, while different sub-activities are pulled apart.

Temporally, we treat a complex activity as a sequence of permutable sub-activities and model the distribution over permutations with a Generalized Mallows Model (GMM) [5]. GMMs have been successfully used in the NLP

¹Here, we refer to the NLP definition of script as "*a predetermined, stereotyped sequence of actions that define a well-known situation*" [22].

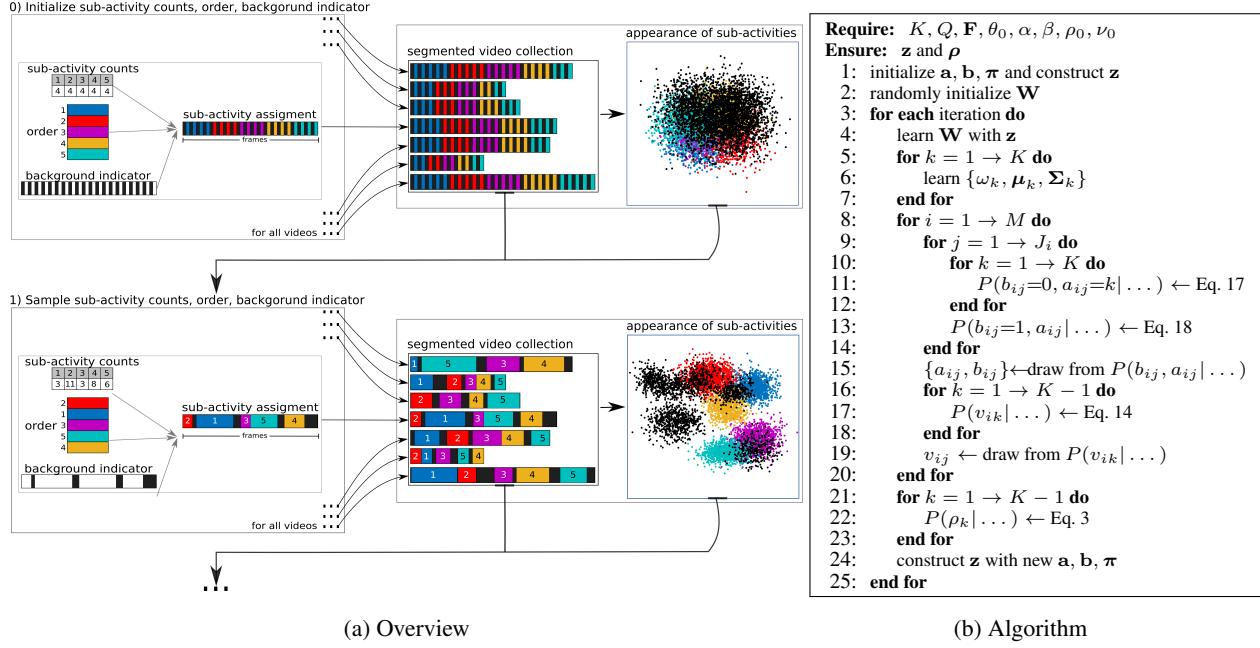


Figure 1: (a) Our iterative model alternates between learning visual appearance and temporal structure of sub-activities. We combine visual appearance with a temporal model to obtain a segmentation of video sequences which is then used to update the visual appearance representation for the next iteration. (b) Algorithm for our model. (Figure best viewed in color.)

community to model document structures [3] and script knowledge [7]. In our method, the GMM assumes that a canonical sequence ordering is shared among videos of the same complex activity. There are several advantages of using the GMM for modelling temporal structure. First and foremost, the canonical ordering enforces a global ordering constraint over the activity – something not possible with Markovian models [12, 19, 23] and recurrent neural networks (RNNs) [32]. Secondly, considering temporal structure as a permutation offers flexibility and richness in modelling. We can allow for missing steps and deviations, all of which are characteristic of complex activities, but cannot be accounted for with works which enforce a strict ordering [1]. Finally, the GMM is compact – parameters grow linearly with the number of sub-activities, versus quadratic growth in pairwise relationships, *e.g.* in HMMs.

Within a video, it is unlikely that every frame corresponds to a specified sub-activity; they may be interspersed with unrelated segments of actors talking or highlighting previous or subsequent sub-activities. Depending on how the video is made, such segments can occur arbitrarily. It becomes difficult to maintain a consistent temporal model under these uncertainties, which in turn affects the quality of visual representations. In this paper we extend our segmentation method to explicitly learn about and represent such “background frames” so that we can exclude them from the temporal model. To summarize our contributions:

- We are the first to explore a fully unsupervised method for temporal understanding of complex activities in

video without requiring any text. We design a discriminative appearance learning model to enable the use of GMMs on state-of-the-art visual features [21, 29, 30].

- We verify our method on real-world videos of complex activities which do not follow strict orderings and are heavily interspersed with background frames.
- We demonstrate that our method achieves competitive results comparable to or better than the state of the art on two challenging complex activity datasets, Breakfast Actions [12] and Inria Instructional Videos [1].

2. Related Work

Modelling temporal structures in activities has been focused predominantly at a frame-wise level [4, 24, 28, 29]. Existing works on complex activity understanding typically require fully annotated video sequences with start and end points of each sub-activity [12, 18, 20]. Annotating every frame in videos is expensive and makes it difficult to work at a large scale. Instead of annotations, a second line of work tries to use cues from accompanying narrations [1, 17, 23]. These works assume that the narrative text is well-aligned with the visual data, with performance governed largely by the quality of the alignment. For example, in the work of Alayrac *et al.* [1], instruction narrations are used as temporal boundaries of sub-activities for discriminative clustering. Sener *et al.* [23], represent every frame as a concatenated histogram of text and visual words, which are used as input to a probabilistic model. The applicability of these

methods is limited because neither the existence of accompanying text, nor their proper alignment to the visual data can be taken for granted.

More recent works focus on developing weakly-supervised solutions, *i.e.* where the orderings of the sub-activities are provided either only during training [9, 19] or testing as well [2]. These methods try to align the frames to the given ordered sub-activities. Similar to us, the work of Bojanowski *et al.* [2] includes a “*background*” class. However, they assume that the background appears only once between every consecutive pair of sub-activities, while our model does not force any constraints on the occurrence of background. Others [9, 19] borrow temporal modelling methods from speech recognition such as connectionist temporal classification, RNNs and HMMs.

In the bigger scope of temporal sequences, several previous works have also addressed unsupervised segmentation [6, 11, 33]. Similar to us in spirit is the work of Fox *et al.* [6], which proposes a Bayesian nonparametric approach to model multiple sets of time series data concurrently. However, it has been applied only to motion capture data. Since skeleton poses are lower-dimensional and exhibit much less variance than video, it is unlikely for such a model to be directly applicable to video without a strong discriminative appearance model. To our knowledge, we are the first to tackle the problem of complex activity segmentation working solely with visual data without any supervision.

3. The Generalized Mallows Model (GMM)

The GMM models distributions over orderings or permutations. In the standard Mallows model [16], the probability of observing some ordering π is defined by a dispersion parameter ρ and a canonical ordering σ ,

$$P_{\text{MM}}(\pi|\sigma, \rho) = \frac{e^{-\rho \cdot d(\pi, \sigma)}}{\psi(\rho)}, \quad (1)$$

where any distance metric for rankings or orderings can be used for $d(\cdot, \cdot)$. The extent to which the probability decreases as π differs from σ is controlled by a dispersion parameter $\rho > 0$; $\psi(\rho)$ serves as a normalization constant.

The GMM, first introduced by Fligner and Verducci [5], extends the standard Mallows model by introducing a set of dispersion parameters $\rho = [\rho_1, \dots, \rho_{K-1}]$, to allow individual parameterization of the K elements in the ordering. The GMM represents permutations as a vector of inversion counts $\mathbf{v} = [v_1, \dots, v_{K-1}]$ with respect to an identity permutation $(1, \dots, K)$, where element v_k corresponds to the total number of elements in $(k+1, \dots, K)$ that are ranked before k in the ordering π ². If we assume that σ is the identity

²Only $K - 1$ elements are needed since v_K is 0 by definition as there cannot be any elements greater than K .

permutation, then a natural distance $d(\pi, \sigma)$ can be defined as $\sum_k \rho_k v_k$, leading to

$$P_{\text{GMM}}(\mathbf{v}|\rho) = \frac{e^{-\sum_k \rho_k v_k}}{\psi_k(\rho)} = \prod_{k=1}^{K-1} \frac{e^{-\rho_k v_k}}{\psi_k(\rho_k)}, \quad (2)$$

with $\psi_k(\rho_k) = \frac{1-e^{-(K-k+1)\rho_k}}{1-e^{-\rho_k}}$ as the normalization.

As the GMM is an exponential distribution, the natural prior for each element ρ_k is the conjugate:

$$P_{\text{GMM}_0}(\rho_k|v_{k,0}, \nu_0) \propto e^{-\rho_k v_{k,0} - \log(\psi_k(\rho_k))\nu_0}, \quad (3)$$

with hyper-parameters $v_{k,0}$ and ν_0 . Intuitively, the prior states that over ν_0 previous trials, $\nu_0 \cdot v_{k,0}$ inversions will be observed [3]. For simplicity, we do not set multiple priors for each k and use a common prior ρ_0 as per [3], such that

$$v_{k,0} = \frac{1}{e^{\rho_0}-1} - \frac{K-k+1}{e^{(K-k+1)\rho_0}-1}. \quad (4)$$

4. Proposed Model

Assume we are given a collection of M videos, all of the same complex activity, and that each video is composed of an ordered sequence of multiple sub-activities. A single video i with J_i frames can be represented by a design matrix of features $\mathbf{F}_i \in \mathbb{R}^{J_i \times D}$, where D is the feature dimension. We further define \mathbf{F} as the concatenated design matrix of features from all M videos and $\mathbf{F}_{\setminus i}$ as the features excluding video i . We first describe how we discriminatively learn the features \mathbf{F} in Sec. 4.1 before describing the standard temporal model in Sec. 4.2 and the full model which models background frames in Sec. 4.3.

4.1. Sub-Activity Visual Features

Within a video collection of a complex activity there may be huge variations in visual appearance, even with state-of-the-art visual feature descriptors [21, 29, 30]. Suppose for frame j of video i we have video features X_{ij} with dimensionality V . These features, if clustered naively, are most likely to group together according to video rather than sub-activity. To cluster the features more discriminantly, we learn a linear mapping of these features into a latent embedding space, *i.e.* $\Phi_f(X_{ij}) : \mathbb{R}^V \rightarrow \mathbb{R}^E$. We also define in the latent space K anchor points, with locations determined by a second mapping $\Phi_a(k) : \{1, \dots, K\} \rightarrow \mathbb{R}^E$. More specifically,

$$\Phi_f(X_{ij}) = \mathbf{W}_f X_{ij}, \quad \mathbf{W}_f \in \mathbb{R}^{E \times V} \quad (5)$$

$$\Phi_a(k) = \mathbf{W}_a(k), \quad \mathbf{W}_a \in \mathbb{R}^{E \times K} \quad (6)$$

where \mathbf{W}_f and \mathbf{W}_a are the learned embedding weights and E is the dimensionality of the joint latent space. Here, $\mathbf{W}_a(k)$ is the k -th column of \mathbf{W}_a , which corresponds to the location of anchor k in the latent space. Together, \mathbf{W}_f and \mathbf{W}_a make up the parameter \mathbf{W} . We use the similarity

of the video feature with respect to these anchor points as a visual feature descriptor, *i.e.*

$$\mathbf{F}_{ij} = \mathbf{W}_a^\top \mathbf{W}_f X_{ij}, \quad (7)$$

where $\mathbf{F}_{ij} = [f^1, \dots, f^K]_{ij}$. Each element f_{ij}^k is inversely proportional to the distance between X_{ij} and anchor point k in the latent space. By using K anchor points, this implies that $D = K$.

Our objective in learning the embeddings is to cluster the video features discriminatively. We achieve this by encouraging the X_{ij} belonging to the same sub-activity to cluster closely around a single anchor point while being far away from the other anchor points. If we assign each anchor point to a given sub-activity, then we can learn \mathbf{W} by minimizing a pair-wise ranking loss L , where

$$L = \sum_{i,j}^{M,J_i} \sum_{k=1,k \neq k^*}^K \max[0, f_{ij}^k - f_{ij}^{k^*} + \Delta] + \gamma \|\mathbf{W}\|_2^2. \quad (8)$$

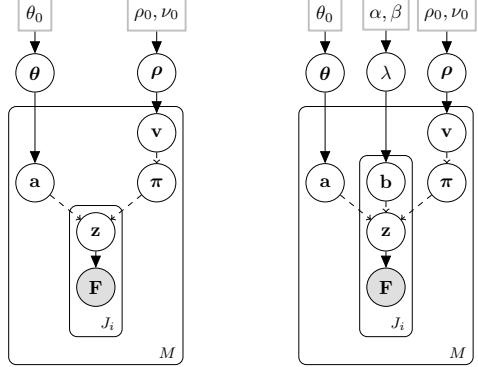
In this loss, k^* is the anchor point associated with the true sub-activity label for \mathbf{F}_{ij} , Δ is a margin parameter and γ is the regularization constant for the l_2 regularizer of \mathbf{W} . The loss in Eq. 8 encourages the distance of X_{ij} in the latent space to be closer to the anchor point k^* associated with the true sub-activity than any other anchor point by a margin Δ .

The above formulation assumes that the right anchor point k^* , *i.e.* the true sub-activity label, is known. This is not the case in an unsupervised scenario so we follow an iterative approach where we learn \mathbf{W} at each iteration from an assumed sub-activity based on the segmentation of the previous iteration. More details are given in Sec. 4.4.

4.2. Standard Temporal Model

Given a collection of M videos of the same complex activity, we would like to infer the sub-activity assignments $\mathbf{z} = \{\mathbf{z}_i\}, i \in \{1, \dots, M\}$. For video i , $\mathbf{z}_i = \{z_{ij}\}, j \in \{1, \dots, J_i\}$, $z_{ij} \in \{1, \dots, K\}$ can be assigned to one of K possible sub-activities³. We introduce \mathbf{a}_i , a bag of sub-activity labels for video i , *i.e.* the collection of elements in \mathbf{z}_i but without consideration for the temporal frame ordering. The ordering is then described by π_i . \mathbf{a}_i is expressed as a vector of counts of the K possible sub-activities, while π_i is expressed as an ordered list. Together, \mathbf{a}_i and π_i determine the sub-activity label assignments \mathbf{z}_i to the frames of video i . (\mathbf{a}, π) are redundant to \mathbf{z} ; the extra set of variables gives us the flexibility to separately model the sub-activities' visual appearance (based on \mathbf{a}) from the temporal ordering (based on π). We model \mathbf{a} as a multinomial, with parameter θ and a Dirichlet prior with hyperparameter θ_0 . For the ordering π , we use a GMM with the exponential prior from Eq. 3 and hyperparameters ρ_0 and ν_0 . The joint distribution of the model factorizes as follows:

³For convenience, we overload the use of K for both the number of elements in the ordering for the GMM as well as the number of sub-activities, as the two are equal when applying the GMM.



(a) Standard model

(b) Full model with background

Figure 2: Plate diagrams of our models. Shaded nodes: observed variables, rectangles: fixed hyper-parameters, dashed arrows: deterministically constructed variables.

$$\begin{aligned} & P(\mathbf{z}, \theta, \rho, \mathbf{F} | \theta_0, \rho_0, \nu_0) \\ &= P(\mathbf{F} | \mathbf{z}) P(\mathbf{a} | \theta) P(\pi | \rho) P(\theta | \theta_0) P(\rho | \rho_0, \nu_0) \\ &= \left[\prod_{i,j=1}^{M,J_i} P(\mathbf{F}_{ij} | z_{ij}) \right] \left[\prod_{i=1}^M P(\mathbf{a}_i | \theta) P(\pi_i | \rho) \right] \\ & \quad \left[\prod_{k=1}^K P(\theta_k | \theta_0) \right] \left[\prod_{k=1}^{K-1} P(\rho_k | \rho_0, \nu_0) \right], \end{aligned} \quad (9)$$

based on the assumption that each frame of each video as well as each video are all independent observations.

We show a diagram of the model in Fig. 2a. When using the GMM, performing MLE to find a consensus or canonical ordering over a set of observed orderings is an NP hard problem, though several approximations have been proposed. Our case is the reverse, in which we assume that a canonical ordering is already given and we would like to find a (latent) set of orderings. Our interest is to infer the posterior $P(\mathbf{z}, \rho | \mathbf{F}, \theta_0, \rho_0, \nu_0)$ for the entire video corpus. Directly working with this posterior is intractable, so we make MCMC sampling-based approximations. Specifically, we use slice sampling for ρ and collapsed Gibbs sampling [8] for \mathbf{z} . Since \mathbf{z} is fully specified by \mathbf{a} and π , it is equivalent to sample \mathbf{a} and π . Before elaborating on the sampling equations, we first detail how we model the video likelihood $P(\mathbf{F}_i | \mathbf{z}_i)$.

Video likelihood $P(\mathbf{F}_i | \mathbf{z}_i)$ can be broken down into the product of frame likelihoods, since each frame is conditionally independent given the frame's sub-activity, *i.e.*

$$P(\mathbf{F}_i | \mathbf{z}_i, \mathbf{F}_{\setminus i}, \mathbf{z}_{\setminus i}) = \prod_{j=1}^{J_i} P(\mathbf{F}_{ij} | z_{ij}, \mathbf{F}_{\setminus i}, \mathbf{z}_{\setminus i}). \quad (10)$$

Since our temporal model is generative, we need to make some assumptions about the generating process behind the video features. We directly model the frame likelihoods and use K mixtures of Gaussians, one for each sub-activity k . Each mixture has Q components with weights ω_k , means

μ_k and covariances Σ_k , with likelihood scores for each mixture selected according to the assignments z_{ij} :

$$P(\mathbf{F}_{ij}|z_{ij} = k, \mathbf{F}_{\setminus i}, \mathbf{z}_{\setminus i}) \sim \sum_{q=1}^Q \omega_k^q \cdot \mathcal{N}(\boldsymbol{\mu}_k^q, \boldsymbol{\Sigma}_k). \quad (11)$$

Sampling sub-activity a_i is done with collapsed Gibbs sampling. Recall that \mathbf{a} is modelled as a multinomial with K outcomes parameterized by θ . We sample a_{ij} , the j -th frame for video i , from the posterior conditioned on all other variables. Without the redundant terms, this posterior is expressed as

$$P(a_{ij} = k | \dots) \propto P(a_{ij} = k | \mathbf{a}_{\setminus ij}, \theta_0) \cdot P(\mathbf{F}_i | \mathbf{z}_i, \mathbf{F}_{\setminus i}, \mathbf{z}_{\setminus i}), \quad (12)$$

where the second term is the video likelihood from Eq. 10. The first term is a prior over the sub-activities, and can be estimated by integrating over θ . The integration is done via the collapsed Gibbs sampling, and, as we assumed that $\theta \sim \text{Dirichlet}(\theta_0)$, this results in

$$P(a_{ij} = k | a_{\setminus ij}, \theta_0) = \frac{N_k + \theta_0}{\sum_{k=1}^K N_k + K\theta_0}, \quad (13)$$

where N_k is the total number of times the sub-activity k is observed in the all sequences and $\sum_{k=1}^K N_k$ is the total number of sub-activity assignments.

Note that sampling a_{ij} does not correspond to the sub-activity assignment to the j -th frame. The assignment is given by z_{ij} , which can only be computed after sampling a_{ij} for all J_i frames of video i and then re-ordering the bag of frames according to π_i .

Sampling ordering π_i is done via regular Gibbs sampling. Recall that the ordering follows a GMM as described in Sec. 3 and is parameterized for elements in the ordering individually via inversion count vector \mathbf{v}_i . As such, we sample a value for each position in the inversion count vector from $k = 1$ to $K - 1$ independently according to

$$P(v_{ik} = c | \mathbf{z}, \rho, \mathbf{F}) \propto P(v_{ik} = c | \rho_k) \cdot P(\mathbf{F}_i | \mathbf{z}_i, \mathbf{F}_{\setminus i}, \mathbf{z}_{\setminus i}), \quad (14)$$

where c indicates the inversion count assignment to v_{ik} . Again, the second term is the video likelihood from Eq. 10, while the first term corresponds to $P_{\text{GMM}}(v_{ik} = c; \rho_k)$, and is computed according to Eq. 2. We estimate the probability of every possible value of v_{ik} , which ranges from 0 to $K - k$, and sample a new inversion count value c based on these probabilities.

Sampling GMM dispersion parameter ρ_k : This is done for each sub-activity $k = 1$ to $K - 1$ independently. We draw ρ_k using slice sampling [15] from the conjugate prior distribution P_{GMM_0} according to Eq. 3.

4.3. Background Modeling

To consider background, we extend the label assignment vector \mathbf{z} with a binary indicator variable $b_{ij} \in \{0, 1\}$ for each frame. The indicator b_{ij} follows a Bernoulli

variable parameterized by λ , with a beta prior, *i.e.* $\lambda \sim \text{Beta}(\alpha, \beta)$. In this setting, \mathbf{z}_i is determined by the bag of sub-activities \mathbf{a}_i , the ordering π_i , and background vector $\mathbf{b}_i = \{b_{ij}\}$, where \mathbf{b}_i indicates the frames to be excluded from sub-activity consideration. For example, for video i , given $\mathbf{a}_i = [6 3 5]$, $\pi_i = [2 3 1]$ and $\mathbf{b}_i = [11100111001100011110011]$, the sub-activity assignment is $\mathbf{z}_i = [22200333003300011110011]$.

We show a diagram of the model in Fig. 2b. The joint distribution of the model can be expressed as

$$\begin{aligned} P(\mathbf{z}, \theta, \rho, \mathbf{F} | \theta_0, \alpha, \beta, \rho_0, \nu_0) &= P(\mathbf{a} | \theta, \theta_0) \\ &\cdot P(\pi | \rho, \rho_0, \nu_0) \cdot P(\mathbf{b} | \lambda, \alpha, \beta) \cdot P(\mathbf{F} | \mathbf{a}, \pi, \mathbf{b}). \end{aligned} \quad (15)$$

Drawing samples from this full model requires a small modification to the sub-activity sampling \mathbf{a}_i . More specifically, we need a blocked collapsed Gibbs sampler that samples a_{ij} and b_{ij} jointly while integrating over θ and λ .

Sampling background \mathbf{b}_i is done from the joint conditional

$$P(b_{ij}, a_{ij} | \dots) \propto P(b_{ij} | \alpha, \beta) \cdot P(a_{ij} | \mathbf{a}_{\setminus ij}, \theta_0) \cdot P(\mathbf{F}_i | \mathbf{z}_i, \mathbf{F}_{\setminus i}, \mathbf{z}_{\setminus i}). \quad (16)$$

This is equivalent to the following for a sub-activity frame:

$$\begin{aligned} P(b_{ij} = 0, a_{ij} = k | \dots) &\propto \frac{N_f + \alpha}{N_f + N_b + \alpha + \beta} \\ &\cdot \frac{N_k + \theta_0}{\sum_{k=1}^K N_k + K\theta_0} \cdot P(\mathbf{F}_i | b_{ij} = 0, a_{ij} = k, \mathbf{F}_{\setminus i}, \mathbf{z}_{\setminus ij}), \end{aligned} \quad (17)$$

where N_f and N_b are the total number of sub-activity frames and background frames in the corpus respectively. For a background frame, the joint conditional is equal to

$$P(b_{ij} = 1, a_{ij} | \dots) \propto \frac{N_b + \alpha}{N_f + N_b + \alpha + \beta} \cdot P(\mathbf{F}_i | b_{ij} = 1, a_{ij}, \mathbf{F}_{\setminus i}, \mathbf{z}_{\setminus ij}). \quad (18)$$

The video likelihood in Eqs. 17 and 18 are computed in a similar way as defined in Eqs. 10 and 11, with the exception that we now iterate over the joint states of background and sub-activity labels for the frame likelihoods. Note that this only adds one extra probability in being computed, *i.e.* $b = 1$, since the state of a_{ij} is then irrelevant. The rest of the Gibbs sampling remains the same.

4.4. Inference Procedure

Our model's inputs are the frames \mathbf{X} , the number of sub-activities K and the number of Gaussian mixtures Q . We iterate between solving for \mathbf{F} and sampling \mathbf{z} and ρ from the posterior $P(\mathbf{z}, \rho | \mathbf{F}, \theta_0, \alpha, \beta, \rho_0, \nu_0)$. To initialize \mathbf{z}_i for each video i , the sub-activity counts \mathbf{a}_i are split uniformly over K sub-activities; π_i is set to the canonical ordering; \mathbf{b}_i is set with every other frame being background (see Fig. 1a). Using the current assignments \mathbf{z} , we first learn \mathbf{W} of the latent embeddings to solve for \mathbf{F} and then for each sub-activity k , the Gaussian mixture components $\{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$. For each video i , we then proceed to re-sample $\{\mathbf{a}_i, \mathbf{b}_i\}, \pi_i$, in that order, using Gibbs sampling to construct \mathbf{z}_i . After

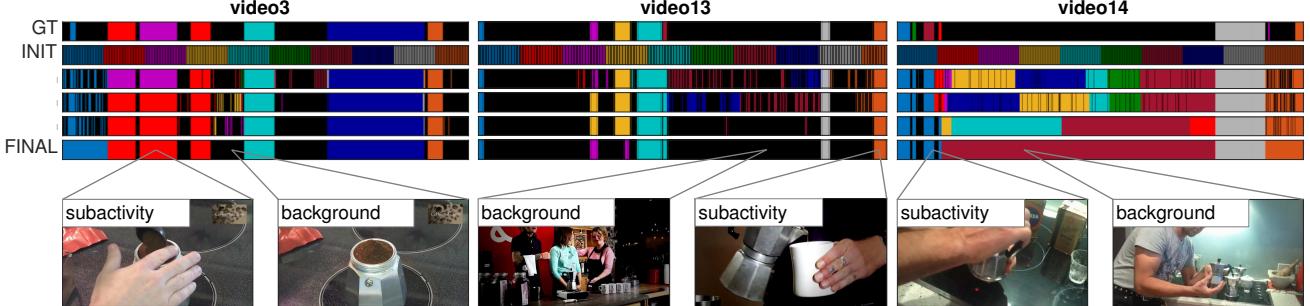


Figure 3: Segmentation outputs on three ‘*making coffee*’ examples from Inria Instructional Videos Dataset [1]. Colors indicate different sub-activities, black the background frames. Since our algorithm is fully unsupervised, we established one-to-one color mappings between the ground truth and our outputs for visualization purposes. The first row (GT) is the ground truth; the remaining rows show the progression from the initialization (INIT) over some iterations to the (FINAL) segmentation. Our method performs well when the appearance of the sub-activities is discriminative, *e.g.* for video 3, occurrence of a hand during a sub-activity vs. none during the background frames, or people talking for video 13. We fail in detecting background when there are also interactions with objects of interest, *e.g.* in video 14. Our model does not enforce continuity over the background frames and may result in fragmentation, but as shown, with good appearance modelling, the background clusters naturally. Furthermore, the final segmentations may contain a different number of sub-activities while still maintaining a global order, *e.g.* the orange sub-activity tends to appear last and follows the grey one.

repeating for each video, we can then re-sample the dispersion parameter ρ . From the new \mathbf{z} and ρ , we then repeat. This process is summarized in the algorithm in Fig. 1b.

To optimize Eq. 8 for learning \mathbf{W} , we use Stochastic Gradient Descent (SGD) with mini-batches of 200 and momentum of 0.9. We set the hyper-parameters $\rho_0 = 1$, $\alpha = 0.2$, $\beta = 0.2$, $\nu_0 = 0.1$, $\theta_0 = 0.1$.

5. Experimentation

5.1. Datasets & Evaluation Metrics

We analyze our model’s performance on two challenging datasets, Breakfast Actions [12] and Inria Instructional Videos [1]. Breakfast Actions has 1,712 videos of 52 participants performing 10 breakfast preparation activities. There are 48 sub-activities, and videos vary according to the participants’ preference of preparation style and orderings. We use the visual features from [13] based on improved dense trajectories [31]. This dataset has no background.

Inria Instructional Videos contains 150 narrated videos of 5 complex activities collected from YouTube. The videos are on average 2 minutes long with 47 sub-activities. We use the visual features provided by [1]: improved dense trajectories and VGG-16 [26] conv5 layer responses taken over multiple windows per frame. The trajectory and CNN features are each encoded with bag-of-words and concatenated for each frame. The videos are labelled, including the background, *i.e.* frames in which the sub-activity is not visually discernible, usually when the person stops to explain past, current or upcoming steps. As such, the sub-activities are separated by hundreds of background frames (73% of all frames). We evaluate our standard model without back-

ground modelling by removing these frames from the sequence as well as our full model on the original sequences.

To evaluate our segmentations in the fully unsupervised setting, we need one-to-one mappings between the segment and ground truth labels. In line with [1, 23], we use the Hungarian method to find the mapping that maximizes the evaluation scores and then evaluate with three metrics: The mean over frames (Mof) evaluates temporal localization of sub-activities and indicates the percentage of frames correctly labelled. The Jaccard index, computed as intersection over detections, as well as the F_1 score quantify differences between ground truth and predicted segmentations. With all three measures, higher values indicate better performance.

We also show a partly supervised baseline in which we use ground truth sub-activity labels for learning \mathbf{F} but learn the temporal alignments unsupervised. This can be thought of as an upper bound on performance for our fully unsupervised version, in which we iteratively learn the temporal alignment and discover the visual appearance of the sub-activities. We refer to these to as “*ours GT*” and “*ours iterated*” respectively in the experimental results.

5.2. Sub-Activity Visual Appearance Modelling

By projecting the frames’ visual features and the sub-activity labels into a joint feature space, we learn a visual appearance model for the sub-activities. We first consider the standard model on Inria Instructional Videos with the background frames removed. The plot in Fig. 4(a) tells us that the appearance model can be learned successfully in an iterative fashion and begins to stabilize after approximately 5 iterations between learning the sub-activity appearance and the GMM. Our model’s performance depend-

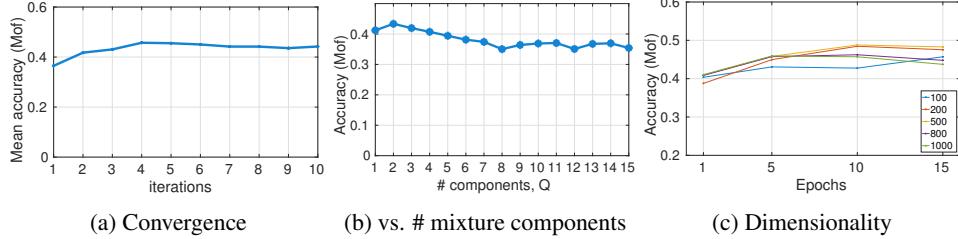


Figure 4: Influence of our model’s parameters are tested on the Instructional Videos Dataset [1] without background frames. We set K to the ground truth sub-activity number of all five activities. Our method’s performance over iterations is shown in (a), using different numbers of Gaussian mixture components in (b) and dimensionality of embedding space in (c).

ing on the the number of Gaussian mixture components Q is shown in Fig. 4(b). The resulting sub-activity representations are very low-dimensional and highly separable so that we achieve higher Mof with a few number of components. We use $Q = 3$ mixture components for our iterative and $Q = 1$ for the ground truth experiments. In Fig. 4(c), we use our iterated method to show the Mof for different values of E , our embedding dimensionality, over the training epochs. We find only small differences in Mof for different E values. We fix the embedding size $E = 200$ with 12 epochs of training and 5 iterations of sub-activity representation and GMM learning for subsequent experiments on both datasets. The run time of a single iteration of our algorithm is proportional to the number of frames J_i in each video and the assumed number of sub-activities K . On a computer with an Intel Core i7 3.30 GHz CPU, our model, for a single iteration, takes approximately 115 seconds (109 for learning the sub-activity appearance model and 6 seconds for estimating the temporal structure).

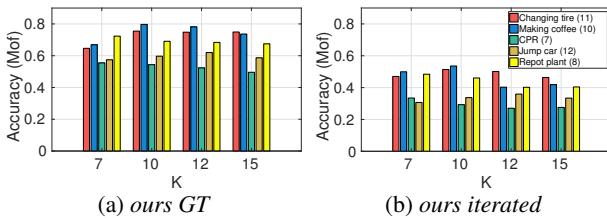


Figure 5: Results (Mof) on Instructional Videos [1] without background frames with varying K . The legend gives the ground truth K for each subactivity in braces.

5.3. Temporal Structure Modelling

The GMM models temporal ordering – without it, one can only classify each frame’s sub-activity label based on the visual appearance. Even if these appearance models are trained on ground truth, the segmentation results would be very poor. On Inria Instructional Videos without background, the average MoF over actions is 0.322 without versus 0.692 with the GMM (see Fig. 5).

The only GMM parameter is K , the number of assumed sub-activities. We again consider Inria Instructional Videos without background and show the MoF as a function of K ,

once partially unsupervised (sub-activity appearance model from ground truth) and once fully unsupervised in Fig. 5(a) and (b) respectively. As can be expected, the MoF drops when moving from the partially to the fully unsupervised case. This drop can be attributed to the fact that the Instructional Videos Dataset is extremely difficult, and exhibits a lot of variation across the videos. In both partially and fully unsupervised cases, however, the MoF remains stable with respect to K , demonstrating that our method is quite robust with respect to varying K . This is also the case once background is considered in the full model with the original sequences (see Fig. 6).

5.4. Background Modelling

In Fig. 7a, we demonstrate the effectiveness of our full model in capturing the background in the original sequences in Inria Instructional Videos. Fig. 7a shows the improvement in MoF once the background is accounted for in the model; there are improvements on every activity, with the most significant being a three-fold increase for ‘jump car’ despite the sequences being 83% background. In Fig. 3, we show qualitative examples of how our model copes with background, where it succeeds, where it fails.

5.5. Comparison to State of the Art

Inria Instructional Videos We compare our full model to [1] in Fig. 6, 7b. The method of [1] outputs a single representative frame for each sub-activity and reports an F1 score on this single frame. To make a valid comparison, since our work is aimed at finding entire segments, we randomly select a frame from each segment and then find a one-to-one mapping based on [14]. Our performance across the five activities is consistent and varies much less than [1]. We have stronger performance in three out of five activities, while we are worse on ‘perform cpr’ and ‘changing tire’. The GMM is a distribution on permutations and orderings; it is by definition unable to account for repeating sub-activities but in ‘perform CPR’, ‘give breath’ and ‘do compression’ are repeated multiple times and account for more than 50% of the sequence frames. In general, we attribute our stronger performance to the fact that the GMM can model flexible sub-activity orderings, while [1] enforces a strict ordering.

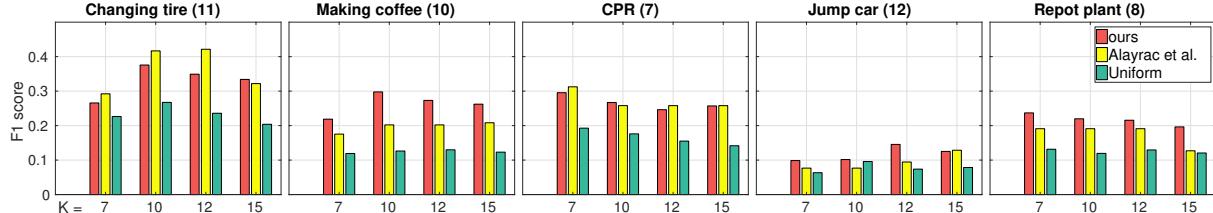


Figure 6: Comparison of our method with Alayrac *et al.* [1] on the Instructional Videos Dataset [1]. To be compatible to the main step detection of Alayrac *et al.* we report the mean over 15 randomly selected frames from each segment.

	Mof	Jaccard
Fully Supervised	SVM [9]	15.8
	HTK [12]	-
Weakly Supervised	OCDC [2]	8.9
	ECTC [9]	27.7
	Fine2Coarse [18]	33.3 47.3
Unsupervised	ours iterated	34.6 47.1

Table 1: Comparisons on Breakfast Actions [12]. Methods are evaluated according to Mof and Jaccard index. For both, a higher result indicates better performance.

The GMM parameter ρ has a prior with hyper-parameter ρ_0 (Sec. 3). A smaller ρ_0 allows more flexible orderings, while a larger ρ_0 encourages the ordering π to remain similar to the canonical ordering σ . In all of our reported results, we fixed $\rho_0 = 1$. We find that for an activity such as ‘*change tire*’, which follows a strict ordering, a larger ρ_0 is more appropriate; with $\rho_0=5$ we are comparable to [1] (0.41 vs. 0.42 F1 score). For ‘*jump car*’ our method outperforms [1], however our overall performance is the lowest as our model struggles with separating the visually very similar ‘*remove cable A*’ and ‘*remove cable B*’.

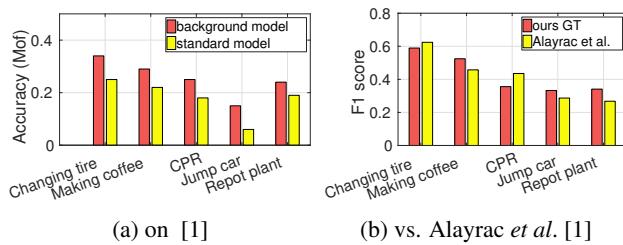


Figure 7: (a) Our standard model vs. background model on original Inria Instructional Videos sequences. The fractions of background are *changing tire* (0.46), *making coffee* (0.71), *perform CPR* (0.56), *jump car* (0.83) and *repot plant* (0.66). (b) Comparison of our supervised setting against Alayrac *et al.*’s supervised method on the Instructional Videos Dataset [1]. Here, our model learns the sub-activity appearance from the ground truth annotations. Alayrac *et al.* use the ground truth annotations as constraints for their discriminative clustering based algorithm.

Breakfast Actions This dataset has no background labels so we apply our standard model and compare with other fully supervised and semi-supervised approaches in Table 1. Of the supervised methods, the SVM method [9] classifies each frame individually without any temporal consideration and achieves an Mof of 15.8%. This shows the strength (and necessity) of temporal information. “*Ours iterated*” is the only fully unsupervised method; we only set K based on ground truth. In comparison, the weakly supervised methods [9, 18, 2] require both K as well as an ordered list of sub-activities as input. ECTC [9] is based on discriminative clustering, while OCDC [2] and Fine2Coarse [18] are both RNN-based methods. We find that our fully unsupervised approach has performance that is state of the art.

6. Conclusion

In this paper we present an unsupervised method for partitioning complex activity videos into coherent segments of sub-activities. We learn a function assigning sub-activity scores to a video frame’s visual features, we model the distribution over sub-activity permutations by a Generalized Mallows Model (GMM). Furthermore, we account for background frames not contributing to the actual activity.

We successfully test our method on two datasets of this challenging problem and are either comparable to or outperform the state of the art, even though our method is completely unsupervised, in contrast to the existing work. Our method is able to produce coherent segments, at the same time being flexible enough to allow missing steps and variations in ordering. Performance drops slightly for complex activities including repetitive sub-activities, as the GMM does not allow for such repeating structures. In the future we plan to investigate approaching this problem in a hierarchical manner to handle repeating blocks as a single step, which can then be further subdivided. Finally, the GMM is unimodal – only one canonical ordering for the set is assumed. This is a valid assumption for activities such as cooking and simple procedural tasks, but we will consider for future work applying multi-modal extensions.

Acknowledgments Research in this paper was supported by the DFG project YA 447/2-1 (DFG Research Unit FOR 2535 Anticipating Human Behavior).

References

- [1] J.-B. Alayrac, P. Bojanowski, N. Agrawal, J. Sivic, I. Laptev, and S. Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *CVPR*, 2016.
- [2] P. Bojanowski, R. Lajugie, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Weakly supervised action labeling in videos under ordering constraints. In *ECCV*, 2014.
- [3] H. Chen, S. Branavan, R. Barzilay, D. R. Karger, et al. Content modeling using latent permutations. *Journal of Artificial Intelligence Research*, 36(1):129–163, 2009.
- [4] B. Fernando, E. Gavves, J. M. Oramas, A. Ghodrati, and T. Tuytelaars. Modeling video evolution for action recognition. In *CVPR*, 2015.
- [5] M. A. Fligner and J. S. Verducci. Distance based ranking models. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 359–369, 1986.
- [6] E. B. Fox, M. C. Hughes, E. B. Sudderth, M. I. Jordan, et al. Joint modeling of multiple time series via the beta process with application to motion capture segmentation. *The Annals of Applied Statistics*, 8(3):1281–1313, 2014.
- [7] L. Frermann, I. Titov, and M. Pinkal. A hierarchical bayesian model for unsupervised induction of script knowledge. In *EACL*, 2014.
- [8] T. L. Griffiths and M. Steyvers. Finding scientific topics. *PNAS*, 101(suppl 1):5228–5235, 2004.
- [9] D.-A. Huang, L. Fei-Fei, and J. C. Niebles. Connectionist temporal modeling for weakly supervised action labeling. In *ECCV*, 2016.
- [10] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.
- [11] B. Krüger, A. Vögele, T. Willig, A. Yao, R. Klein, and A. Weber. Efficient unsupervised temporal segmentation of motion data. *IEEE Transactions on Multimedia (TMM)*, 19(4):797–812, 2017.
- [12] H. Kuehne, A. Arslan, and T. Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *CVPR*, 2014.
- [13] H. Kuehne, J. Gall, and T. Serre. An end-to-end generative framework for video segmentation and recognition. In *WACV*, 2016.
- [14] T. W. Liao. Clustering of time series dataa survey. *Pattern recognition*, 38(11):1857–1874, 2005.
- [15] D. J. MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.
- [16] C. L. Mallows. Non-null ranking models. i. *Biometrika*, 44(1/2):114–130, 1957.
- [17] J. Malmaud, J. Huang, V. Rathod, N. Johnston, A. Rabinovich, and K. Murphy. What’s cookin’? Interpreting cooking videos using text, speech and vision. In *NAACL HLT 2015*, 2015.
- [18] A. Richard and J. Gall. Temporal action detection using a statistical language model. In *CVPR*, 2016.
- [19] A. Richard, H. Kuehne, and J. Gall. Weakly supervised action learning with rnn based fine-to-coarse modeling. In *CVPR*, 2017.
- [20] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *CVPR*, pages 1194–1201, 2012.
- [21] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *IJCV*, 105(3):222–245, 2013.
- [22] R. C. Schank and R. P. Abelson. Scripts, plans, and knowledge. In *IJCAI*, 1975.
- [23] O. Sener, A. R. Zamir, S. Savarese, and A. Saxena. Unsupervised semantic parsing of video collections. In *ICCV*, 2015.
- [24] S. Sharma, R. Kiros, and R. Salakhutdinov. Action recognition using visual attention. *arXiv preprint arXiv:1511.04119*, 2015.
- [25] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.
- [26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.
- [27] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.
- [28] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using lstms. In *ICML*, 2015.
- [29] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.
- [30] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *IJCV*, 103(1):60–79, 2013.
- [31] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.
- [32] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *CVPR*, 2016.
- [33] F. Zhou, F. De la Torre, and J. K. Hodgins. Hierarchical aligned cluster analysis for temporal clustering of human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(3):582–596, 2013.