# Learning to Recognize Human Activities from Soft Labeled Data

Ninghang Hu, Zhongyu Lou, Gwenn Englebienne and Ben Kröse
Informatics Institute, University of Amsterdam, The Netherlands
{n.hu,z.lou,g.englebienne,b.j.a.krose}@uva.nl

***Abstract*—An activity recognition system is a very important component for assistant robots, but training such a system usually requires a large and correctly labeled dataset. Most of the previous works only allow training data to have a single activity label per segment, which is overly restrictive because the labels are not always certain. It is, therefore, desirable to allow multiple labels for ambiguous segments. In this paper, we introduce the method of *soft labeling*, which allows annotators to assign multiple, weighted, labels to data segments. This is useful in many situations, *e.g.* when the labels are uncertain, when part of the labels are missing, or when multiple annotators assign inconsistent labels. We treat the activity recognition task as a sequential labeling problem. Latent variables are embedded to exploit sub-level semantics for better estimation. We propose a novel method for learning model parameters from soft-labeled data in a max-margin framework. The model is evaluated on a challenging dataset (CAD-120), which is captured by a RGB-D sensor mounted on the robot. To simulate the uncertainty in data annotation, we randomly change the labels for transition segments. The results show significant improvement over the state-of-the-art approach.**

## I. Introduction

Activity recognition is an important task for assistant robots, particularly in elderly care [1] (Fig. 1). This topic has been widely studied in both robotics communities [3, 5, 10, 14] and other fields [15, 19, 20]. Most of the work uses a dataset where labels are hard assigned regardless of uncertainty. Learning from these data, however, is often quite problematic because the labeling uncertainty is not captured.

The traditional way of labeling data requires human annotators to watch through the video and choose the *exact* transition point between consecutive activities. This is often quite hard as frames near the transition field tend to be very similar to each other, as is illustrated in Fig. 2. Also, labels of the frames can be too uncertain to be visually discerned. A typical solution to this problem is to play the video back and forth and to check the context for disambiguation. This can reduce the risk of assigning incorrect labels to the sequences, yet it is a rather time-consuming task and the resulting labels can still be erroneous. To be more efficient and also more accurate, the labeling tasks are usually distributed to multiple annotators. As annotating data is a very subjective process, the assigned labels may contradict each other. To deal with this problem, one can pick the most commonly assigned label as the final assignment, thus throwing away the rest of annotations, even though these may still be quite informative.

Fig. 1. Assistant robot in elderly care. Assistant robot is usually equipped with a RGB-D camera that can have both color and depth view. Based on the sensory data, human activities can be recognized and the robot can then provide proper services afterwards. The picture (left) shows after the elderly drinks water, Care-O-bot 3 [13] offers the elderly its tray for placing the cup. The image on the right shows a sample image captured by the depth sensor.

We formulate the task of activity recognition as sequential labeling problem. *i.e.* Knowing a sequence of observations (*e.g.* a RGB-D video), we would like to predict the most likely underlying sequence of activity labels. In our model, we represent activities and video segments as nodes, and we use edges to mode the compatibility among them. Hurried readers may want to see our graphical model in Fig. 3, the details of which will be presented in Section III.

In this paper, we introduce *soft labeling* (Section V-A), a method that allows labeling a single video segment with multiple choices. The name is defined in contrast to the hard assignment of a single label for each video segment. This is very useful when labels of the segments are not certain. For example, the boundary between two activities is usually not very clear, see Fig. 2. Using soft labeling, the annotator can simply choose to assign both labels with 0.5 confidence. Soft labeling can be used to represent partially labeled data as well. For example, for segments where labels are missing or hard to be defined, we can make the assignment uniformly distributed over all the labels. Moreover, when multiple annotators are involved, the soft labels can be encoded as a multinomial distribution according to the voting from multiple annotators. Therefore, the soft labeling is a more flexible way compared with the normal annotation methods.

For learning, we propose a novel loss function that incorporates the soft labeling in a max-margin learning framework (Section V). Unlike the typical zero-one loss, our loss function can give values ranging from zero to one. Compared with the approaches that model labeling uncertainty by adding nodes in the graphical model [18], our method does not increase the computational complexity of the model, as it is independent of the graphical structure. Our source code is available at `http://ninghanghu.eu/activity_recognition.html`.

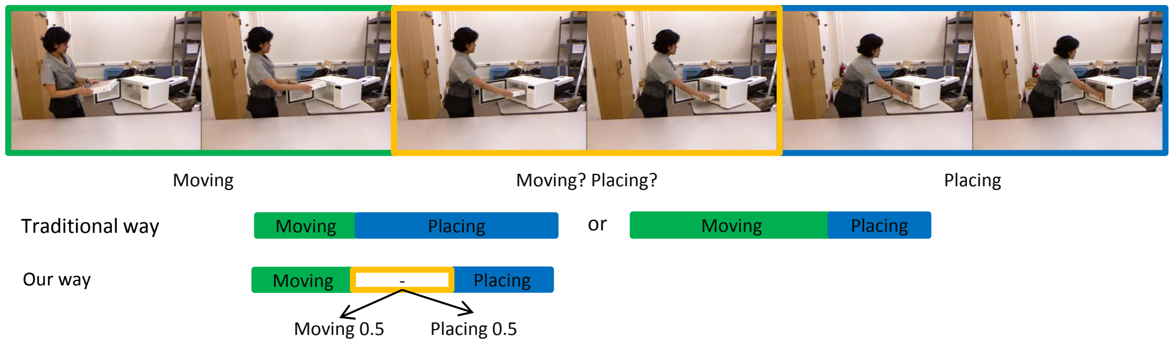


Fig. 2. Comparison between the traditional way of annotation and our way (soft labeling). The video is sampled from the CAD-120 dataset [10]. Each image in the picture represents a video segment. In this example, the subject performs two activities, "moving" (green) and "placeing" (blue). Labels of the first two and last two frames are easy to assign. However, assigning labels for the frames between the two activities is purely based on personal preferences (traditional way). Instead of assigning an arbitrary label, we propose to assign 0.5 to both labels (our way) and let the learning algorithm to determine which label is correct.

## II. Related work

Most prior works in robotics and computer vision formulate the activity recognition problem as Conditional Random Fields (CRFs) [4, 5, 6, 9, 10, 14, 15, 20, 21]. The CRFs model the environment with nodes and edges. The activity labels within such a framework can be interpreted by finding the most likely states which maximize the overall score of the graphical model. As such structure directly models the posterior probability of the activities regardless of the correlation between input features, it provides an easy way for data fusing and thereby fits naturally to robotic scenarios where usually multiple types of sensors are facilitated. A variety of graphical models have been proposed in the literature. Vail et al. [19] model the activities in a linear-chain structure, where activity nodes of the temporal segments are inter-connected. Koppula et al. [10] model both activities and objects affordance as random variables. These nodes are inter-connected to model object-object and object-human interactions. Nodes are connected across the segments to enable temporal interaction. Given a test video, the model jointly estimates both human activities and object affordance labels using a graph-cut algorithm. Multiple segmentation hypotheses are considered when predicting activities, and a two-step learning algorithm is applied to combine predictions from different segmentation hypotheses. Koppula and Saxena [8] also combine multiple segmentation hypotheses by majority voting, which makes it possible to express the uncertainty over labels at test time but still ties the model to use the provided labels for training. Hu et al. [5] encode the interactions between objects and humans at the feature level. In addition, they propose to add a latent layer to exploit underlying semantics between temporal segments. The latent variables, activity variables, and input features within the same segments are fully connected to avoid making inappropriate conditional independence assumptions. The inference algorithm of their model is very efficient as the graph can be viewed as a linear-chain structure. Compared with [10], they show significantly improved performance on a challenging dataset [10] where the labels are fully annotated. However, both of their models fail to capture the uncertainty of the labels for training the model. In contrast, our method explicitly incorporates the uncertainty over the labels at training time, and it is also very flexible in data labeling, *i.e.* the temporal segment can have a single label, multiple labels, or even a missing label.

The learning algorithm of our model is closely related to recent work in the machine learning community. Parameters of the CRFs are usually learned using the Structured SVMs. Tsochantaridis and Hofmann [16] are among the first to generalize standard SVMs into Structured SVMs. The Structured SVMs can give predictions to a set of random variables that follow a certain structure. The model parameters can be learned efficiently using the cutting plane algorithm. Yu and Joachims [24] extend the Structured SVM framework to allow for latent variables (Latent Structured-SVM) and show that the model with latent variables has the capacity to model more complex data. Their model parameters are learned with the Concave Convex Procedure (CCCP), which is an EM-like algorithm that iteratively estimates model parameters and latent variables. Recently, Lou and Hamprecht [11] extend the latent Structured-SVM to allow for learning with partially annotated data. Their learning approach treats the missing labels as latent variables, and the parameters are learned in a similar way as in [24]. The model is evaluated in a tracking task where only part of the detections have been associated with tracks. In their framework, annotations of the training data can either be a single label or completely missing. Assigning a single label is problematic as data may not be distinctive enough for hard assignment. In contrast, we propose a novel learning method that allows for soft-labeled data which contain labeling uncertainty.

## III. Model

Our graphical model is illustrated in Fig. 3. Let $\mathbf{x} = \{x_1, x_2, \ldots, x_K\}$ be the sequence of observations. Based on these observations, we would like to predict the most likely underlying activities $\mathbf{y} = \{y_1, y_2, \ldots, y_K\}$. We write $\mathbf{z} = \{z_1, z_2, \ldots, z_K\}$ as the latent variables in the model. Note that during inference, we use $\mathbf{y}$ to denote activity variables that are estimated. During training, we make distinction between fixed labels $\mathbf{y}$ and uncertain labels $\mathbf{h}$. See Table I for a summary of the notations that are used in this paper.

TABLE I
SUMMARY OF NOTATIONS

| Symbols | Definition |
|---|---|
| $\mathbf{x}$ | Input observations |
| $\mathbf{y}$ | Activity labels (observed during training) |
| $\mathbf{h}$ | Activity labels that are uncertain |
| $\mathbf{z}$ | Latent variables |
| $\boldsymbol{\pi}$ | Soft labels |
| $\hat{\mathbf{y}}, \hat{\mathbf{z}}$ | Predictions |
| $N$ | Total number of training examples |
| $K$ | Length of the training example |
| $D$ | Dimension of input features |
| $M$ | Cardinality of activities nodes |
| $L$ | Cardinality of latent variables |
| $\psi$ | Potential |
| $\phi(x)$ | Feature mapping function |
| $\mathbf{w}$ | Parameter vector |
| $\mathbf{w}(y, z)$ | Parameters that are associated with $y$ and $z$ |

Each observation $x_k$ itself is a feature vector extracted from the segment $k$. The form of $x_k$ is quite flexible. It can be collections of data from different sources, *e.g.* simple sensor readings, human locations, human pose, object locations. Some of these observations may be highly correlated with each other, *e.g.* wearable accelerometers and motion sensors. Thanks to the discriminative nature of our model, we do not need to model such correlation among the observations.

## A. Objective Function

Our model contains three types of potentials that jointly form the objective function.

The first potential measures the score of seeing an observation $x_k$ with a joint-state assignment $(z_k, y_k)$. We define $\phi(x_k)$ to be the function that maps the input data into the feature space. $\mathbf{w}_1 \in \mathbb{R}^{M\times L\times D}$, $\mathbf{w}_2 \in \mathbb{R}^{M\times L}$ and $\mathbf{w}_3 \in \mathbb{R}^{M^2\times L^2}$ are the model parameters. We compute the first potential as

$$\psi_1(y_k, z_k, x_k; \mathbf{w}_1) = \mathbf{w}_1(y_k, z_k) \cdot \phi(x_k) \tag{1}$$

where the $\mathbf{w}_1(y_k, z_k)$ selects the parameters that associated with $y_k$ and $z_k$.

This potential models the full connectivity among $y_k$, $z_k$ and $x_k$, avoiding making any conditional independence assumption. It is more accurate to have such a structure since $z_k$ and $x_k$ may not be conditionally independent over a given $y_k$ in many cases. For example, one could imagine that $y_k$ refers to the activity "opening". However, "opening" is a rather abstract concept that can be associated with many subtypes, such as opening a bottle ($z_k = 1$), opening a microwave ($z_k = 2$), and opening a fridge door ($z_k = 3$). Although all these subtypes belong to "opening", their input features may vary largely. Therefore we use the latent variable $z_k$ to exploit the sub-level semantics.

The second potential measures the score of coupling $y_k$ with $z_k$. It can be considered as either the bias entry of Eq. (1) or the prior of seeing the joint state $(y_k, z_k)$.

$$\psi_2(y_k, z_k; \mathbf{w}_2) = \mathbf{w}_2(y_k, z_k) \tag{2}$$

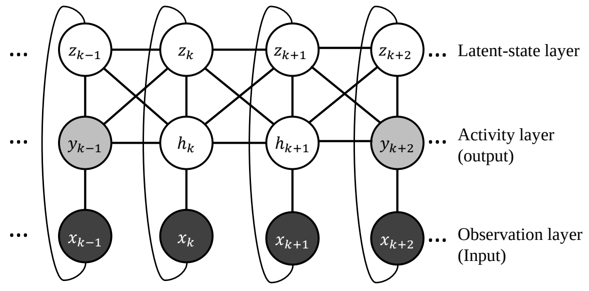


Fig. 3. The graphical representation of our model. Input nodes $\mathbf{x}$ (black) are observed during both training and testing. $\mathbf{y}$ and $\mathbf{h}$ are both the target labels to be predicted. The difference is that $\mathbf{y}$ (gray) is fixed during training, while $\mathbf{h}$ (white) is uncertain. The uncertain labels $\mathbf{h}$ are treated the same as the latent variables $\mathbf{z}$. Note that $\mathbf{x_k}$, $\mathbf{y_k}$, $\mathbf{z_k}$ are fully connected within segments, and the same for the nodes connecting consecutive segments.

The third potential characterizes the transition score from the joint state $(y_{k-1}, z_{k-1})$ to $(y_k, z_k)$. Comparing with the normal transition potentials [22], our model leverages the latent variable $z_k$ for modeling richer contextual information over consecutive temporal segments. Not only does our model contain the transition between states $y_k$, but it also captures the sub-level context using the latent variables. Intuitively, our model is able to capture the fact that the subtype of an activity is more likely to be preceded by a certain subtype of the other activity.

$$\psi_3(y_{k-1}, z_{k-1}, y_k, z_k; \mathbf{w}_3) = \mathbf{w}_3(y_{k-1}, z_{k-1}, y_k, z_k) \tag{3}$$

Summing all potentials over the whole sequence, we can write the objective function of our model as follows

$$\begin{aligned} F(\mathbf{y}, \mathbf{z}, \mathbf{x}; \mathbf{w}) = &\sum_{k=1}^{K} \{\mathbf{w}_1(y_k, z_k) \cdot \phi(x_k) + \mathbf{w}_2(y_k, z_k)\} \\ &+ \sum_{k=2}^{K} \mathbf{w}_3(y_{k-1}, z_{k-1}, y_k, z_k) \end{aligned} \tag{4}$$

The objective function evaluates the matching score between the joint states $(\mathbf{y}, \mathbf{z})$ and the input $\mathbf{x}$. The score can be seen as the unnormalized joint probability in the log space. The objective function can be rewritten into a more general linear form $F(\mathbf{y}, \mathbf{z}, \mathbf{x}; \mathbf{w}) = \mathbf{w} \cdot \Phi(\mathbf{y}, \mathbf{z}, \mathbf{x})$. Therefore the model is in the class of log-linear models.

Note that it is not necessary to specify the role of the latent variables explicitly, but rather the latent variables can be learned automatically from the training data. Theoretically, the latent variables can represent any form of data, *e.g.* time duration, action primitives, as long as it can help with solving the task.

One may notice that our graphical model has many loops, which in general makes exact inference intractable. Since our graph complies with the semi-Markov property, next, we will show that how we benefit from such a structure for efficient inference and learning.

## IV. Inference

Given the graph, the model parameters and the input sequence $\mathbf{x}^{(i)}$, the goal of inference is to find the most likely joint states $\hat{\mathbf{y}}$ and $\hat{\mathbf{z}}$ that maximize the objective function.

$$(\hat{\mathbf{y}}, \hat{\mathbf{z}}) = \underset{\mathbf{y}\in\mathcal{Y},\mathbf{z}\in\mathcal{Z}}{\operatorname{argmax}} F(\mathbf{y}, \mathbf{z}, \mathbf{x}^{(i)}; \mathbf{w}) \tag{5}$$

Generally, solving Eq. (5) is an NP-hard problem that requires evaluating the objective function over an exponential number of state sequences. Exact inference is usually preferable as it is guaranteed to find the global optimum. However, exact inference usually can only be applied efficiently when the graph is acyclic. In contrast, approximate inference is more suitable for loopy graphs, but may take longer to converge and is likely to find a local optimum. Although our graph contains loops, we note that we can transform the graph into a linear-chain structure, in which the exact inference becomes tractable. If we collapse the latent variable $z_k$ with the target variable $y_k$ into a single node, the edges between $z_k$ and $y_k$ become the internal factor of the new node and the transition edges collapse into a single transition edge. This results in a typical linear-chain CRF, where the cardinality of the new nodes is $M \times L$. In the linear-chain CRF, exact inference can be performed efficiently using the dynamic programming.

$$\begin{aligned}V_k(y_k, z_k) =& \mathbf{w}_1(y_k, z_k) \cdot \phi(x_k) + \mathbf{w}_2(y_k, z_k) + \\ & \max_{(y_{k-1},z_{k-1})\in\mathcal{Y}\times\mathcal{Z}} \{\mathbf{w}_3(y_{k-1}, z_{k-1}, y_k, z_k) \\ & + V_{k-1}(y_{k-1}, z_{k-1})\}\end{aligned} \tag{6}$$

Computing Eq. (6) once involves $O(ML)$ computations. In total, Eq. (6) needs to be evaluated for all possible assignment of $(y_k, z_k)$, so that it is computed $ML$ times. The total computational cost is, therefore, $O(M^2 L^2 K)$. Such computation is manageable when $ML$ is not very large, which is usually the case for the task of activity recognition. In our implementation, we use LibDAI [12] as the inference engine.

## V. Learning

We use a max-margin formulation for learning the model parameters, see Algorithm 1. Different from the normal approach, we do not use hard assignment of labels, but rather we propose to use soft labels which contain labeling uncertainty.

### A. Soft Labeling

Assume there are $N$ labeled training examples $(\mathbf{x}^{(1)},\boldsymbol{\pi}^{(1)})$, …, $(\mathbf{x}^{(N)},\boldsymbol{\pi}^{(N)})$, where $\mathbf{x}$ represents a sequence of observations and $\boldsymbol{\pi}$ denotes soft labeling vector. Each training example is a video that is formed with $K$ temporal segments, where $K$ may vary over different training examples. We denote $\pi_k$ as a soft labeling vector that models the possibility of all activities. Specifically, $\pi_k$ is a vector of non-negative values which satisfy

$$\sum_{y\in\mathcal{Y}} \pi_k(y) = 1 \tag{7}$$

Intuitively, $\pi_k$ can be viewed as multinomial distribution over activity labels of the video segment $k$, *i.e.* $\pi_k(a)$ models the probability of the $k^{th}$ segment belonging to the activity $a$.

The soft-labeled data can be obtained very efficiently. After annotation, we collect the labels for each segment, and let us denote the label collection of segment $k$ as $A_k$. Note that the same labels may occur multiple times in the collection if the labels are given by multiple annotators. We compute the soft-label vector $\pi_k$ as

$$\pi_k(y) = \begin{cases} q_y/p & \text{if } p > 0 \\ 1/M & \text{if } p = 0 \end{cases} \tag{8}$$

where $M$ is the number of activities to be recognized, $q_y$ counts the occurrence of the label $y$ in $A_k$, and $p$ is the total number of labels in the collection, *i.e.* $p = \sum_i q_i$.

The soft labeling can be considered as a generalized representation of many other tasks. By adding additional constraints to Eq. (8), our method can be changed into three special cases.

- When $p = 1$, only one label is allowed in $A_k$. This is equivalent to the traditional labeling method (*e.g.* [5]), where labels are hard assigned.
- When $p = 0$, there is no label in collection $A_k$, and all labels obtain the same value $1/M$. This is equivalent to an unsupervised learning problem (*e.g.* [23]) where no label is given.
- When $p \in \{0, 1\}$, part of the labels are given and part of the labels can be missing, which is a typical problem of learning with partially annotated data (*e.g.* [11]).

Benefiting from such a general representation, soft labeling is a very flexible method for data annotation. For the labels that are very certain, one can just assign a single label. For the transition segment, labels are likely to be the confused with its two adjacent segments. Thus both of the labels can be selected. For data that are very ambiguous, the label can be left as empty. In addition to the high flexibility, soft labeling is also very useful when the data are labeled by multiple annotators. With soft labeling, we can keep the uncertainty of the labeling without throwing away any annotation.

**Algorithm 1** Learning parameters with soft-labeled data

```
Input: $N$ labeled training examples $(\mathbf{x}^{(1)},\boldsymbol{\pi}^{(1)})$, $(\mathbf{x}^{(2)},\boldsymbol{\pi}^{(2)})$, …, $(\mathbf{x}^{(N)},\boldsymbol{\pi}^{(N)})$
Set $t = 0$, $s_0 = +\infty$
Set activity nodes to $\mathbf{h}$ or $\mathbf{y}$ based on $\boldsymbol{\pi}$
Initialize $\mathbf{h}^*$ and $\mathbf{z}^*$
repeat
    Set $t = t + 1$
    $f(\mathbf{w}) = \frac{\|\mathbf{w}\|^2}{2} + C\sum \max \Delta(\boldsymbol{\pi}^{(i)}, \mathbf{y}) + F((\mathbf{y}, \mathbf{h}), \mathbf{z}, \mathbf{x}^{(i)})$
    $g(\mathbf{w}) = C\sum_{i=1}^{N} F(\mathbf{y}^{(i)}, \mathbf{h}^*, \mathbf{z}^*, \mathbf{x}^{(i)}; \mathbf{w})$
    Solve $\mathbf{w}_t = \operatorname{argmin}_{\mathbf{w}} f(\mathbf{w}) - g(\mathbf{w})$
    for $i = 1$ to $N$ do
        $(\mathbf{h}^*, \mathbf{z}^*) = \operatorname{argmax}_{\mathbf{h}\in\mathcal{Y},\mathbf{z}\in\mathcal{Z}} F((\mathbf{y}^{(i)}, \mathbf{h}), \mathbf{z}, \mathbf{x}^{(i)}; \mathbf{w}_t)$
    end for
    update $\mathbf{h}^*$, $\mathbf{z}^*$ in $g(\mathbf{w})$
    $s_t = f(\mathbf{w}_t) - g(\mathbf{w}_t)$
until $s_{t-1} - s_t < thres$
Output: $\mathbf{w}_t$
```

### B. Learning Parameters with Soft Labels

This section introduces how we incorporate soft labeling into a max-margin learning framework. For each $\pi_k$ we use a threshold $\delta$ to separate the activity variable into two groups, labels that are known $\mathbf{y}$ and the ones that are uncertain $\mathbf{h}$. If the maximum of $\pi_k$ is larger than $\delta$, it refers that the annotator is highly confident with the label, therefore we treat these labels as known. These known labels are fixed during learning. If the maximum is below the threshold, we consider the label as uncertain, and we treat these activity variables as latent nodes.

The task of learning is to solve the following optimization problem.

$$\min_{\mathbf{w}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \Delta(\boldsymbol{\pi}^{(i)}, \hat{\mathbf{y}}) \right\} \tag{9}$$

where $C$ is a regularization constant and $\hat{\mathbf{y}}$ is predicted by Eq. (5). $\|\mathbf{w}\|^2$ is a regularization term to prevent overfitting. $\Delta(\boldsymbol{\pi}^{(i)}, \hat{\mathbf{y}})$ is a loss function that measures the distance between the annotation and the predicted labels.

We define a novel loss function as follows

$$\Delta(\boldsymbol{\pi}^{(i)}, \hat{\mathbf{y}}) = \sum_{k=1}^{K} 1 - \pi_k^{(i)}(\hat{y}_k) \tag{10}$$

In contrast to the zero-one loss which are commonly used in related work, our loss function allows for soft assignment to the labels and it is capable of encoding the uncertainty of the labels. The zero-one loss can be viewed as a special case when $\pi \in \{0, 1\}$.

Optimizing Eq. (9) directly is not possible as the loss function is non-differentiable. Following [17] and [24], we use the Margin Rescaling Surrogate that serves as an upper bound of the loss function.

After substituting the loss function by the surrogate, the objective function is a summation between a convex term and a concave term. This can be solved with the Concave-Convex Procedure (CCCP) [25]. By substituting the concave function with its tangent hyperplane , the concave term is converted into a function that is linear over the parameters.

$$\min_{\mathbf{w}} \{ \underbrace{\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \max_{\substack{\mathbf{y},\mathbf{h}\in\mathcal{Y}\\ \mathbf{z}\in\mathcal{Z}}} [\Delta(\boldsymbol{\pi}^{(i)}, \mathbf{y}) + F((\mathbf{y},\mathbf{h}),\mathbf{z},\mathbf{x}^{(i)};\mathbf{w})]}_{\text{convex function}} \underbrace{-C\sum_{i=1}^{N} F((\mathbf{y}^{(i)},\mathbf{h}^*),\mathbf{z}^*,\mathbf{x}^{(i)};\mathbf{w})}_{\text{linear function}} \tag{11}$$

where

$$(\mathbf{h}^*, \mathbf{z}^*) = \operatorname*{argmax}_{\mathbf{h}\in\mathcal{Y},\mathbf{z}\in\mathcal{Z}} F((\mathbf{y}^{(i)},\mathbf{h}),\mathbf{z},\mathbf{x}^{(i)};\mathbf{w}_{\text{old}}) \tag{12}$$

and $\mathbf{w}_{\text{old}}$ is the model parameter from the previous iteration.

The inference problem of Eq. (12) can be solved in a similar way as Eq. (5). The difference is that in Eq. (12) part of the labels are observed. We can add the observed labels as the evidence into the graphical model, thereby the same inference algorithm can be applied for finding the latent states.

We can rewrite Eq. (11) in the form of minimizing a function subject to a set of constraints by introducing slack variables:

$$\min_{\mathbf{w},\xi} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \xi_i \right\} \tag{13}$$
$$s.t.\ \forall i \in \{1, 2, \ldots, N\}:$$
$$F((\mathbf{y}^{(i)},\mathbf{h}^*),\mathbf{z}^*,\mathbf{x}^{(i)};\mathbf{w}) - F((\hat{\mathbf{y}},\hat{\mathbf{h}}),\hat{\mathbf{z}},\mathbf{x}^{(i)};\mathbf{w}) \geq \Delta(\boldsymbol{\pi}^{(i)}, \hat{\mathbf{y}}) - \xi_i$$

where the most violated constraints can be computed using augmented inference.

$$(\hat{\mathbf{y}}, \hat{\mathbf{h}}, \hat{\mathbf{z}}) = \operatorname*{argmax}_{\mathbf{y},\mathbf{h}\in\mathcal{Y},\mathbf{z}\in\mathcal{Z}} [\Delta(\boldsymbol{\pi}^{(i)}, \mathbf{y}) + F((\mathbf{y},\mathbf{h}),\mathbf{z},\mathbf{x}^{(i)};\mathbf{w})] \tag{14}$$

The above augmented inference can be solved in a similar way as Eq. (5). We can plug the loss functions as extra factors into the graph and attach them with the *known* target variables. The unobserved target variables are treated the same as the latent variables during augmented inference.

This transforms the optimization problem into a standard Structured SVM form, which has been solved in [17].

Another intuitive way to understand our algorithm is to consider it as solving the learning problem with incomplete data using Expectation-Maximization (EM). In our training data, both the latent variables and part of the ground-truth labels are uncertain. We can start by initializing the latent variables and the missing labels. Once we have done that, the data become complete. Then we can use the standard Structured-SVM to learn the model parameters (similar to the M-step). After that, we can update the latent states again using the parameters that are learned (as in the E-step). The iteration continues until convergence, see Algorithm 1 for more details.

The CCCP algorithm decreases Eq. (9) in every iteration. However, it cannot guarantee finding the global optimum. To avoid being trapped in the local minimum, the latent variables need to be carefully initialized. We use a data-driven approach for the initialization, and details will be presented in Section VI-D.

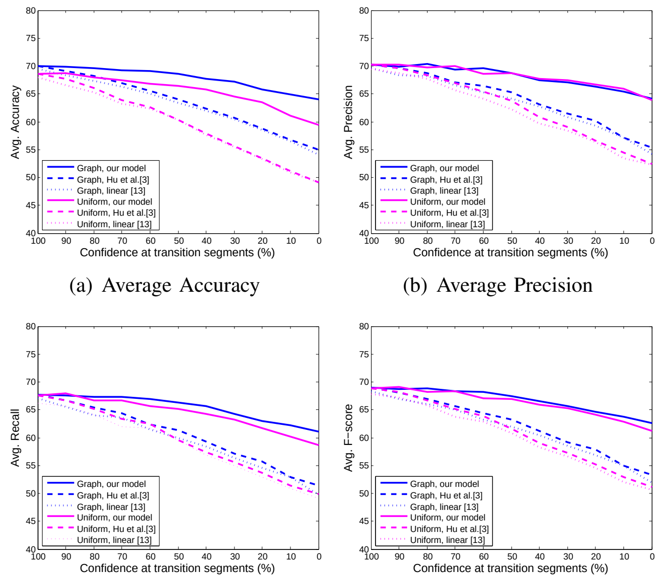
Note that the inference algorithm is extensively used in learning. As we are able to compute the exact inference by transforming the loopy graph into a linear-chain graph, our learning algorithm is much faster and more accurate compared with the other approaches with approximate inference.

## VI. Experiments and Results

Our experiments investigate the following three questions: a) how detrimental are hard assigned labels, *i.e.* each segment can only be associated with a single label when the labels are uncertain. b) How much can we benefit from soft labeling compared with the hard assignment. c) What are the effects of using latent variables.

### A. Experiment Setup

We start the experiments by assuming labels from the CAD-120 dataset are perfectly annotated, *i.e.* annotators are 100% sure about their labeling, and they never make mistakes. In practice, it is impossible to have undoubted labeling, especially for the transition segments where labels are mostly very uncertain. When the annotators are less confident of which label



(a) Average Accuracy (b) Average Precision

(c) Average Recall (d) Average F1-score

Fig. 4. The test performance of activity recognition on the CAD-120 dataset. We show the change of performance with the increasing amount of noise at transition labeling. The colors encode the two different segmentation methods, *i.e.* uniform segmentation (purple) and graph-based segmentation (blue). The type of the lines is used to distinguish three approaches, *i.e.* the baseline (dotted line), the state-of-the-art (dashed line) and the proposed soft labeling approach (solid line). We report the performance in accuracy (a), precision (b), recall (c), and F1-score (d).

to assign, they are more likely to make mistakes. Therefore we simulate the uncertainty of labeling by randomly flip the label of transition segments to the label of the adjacent segment, with certain probability which we vary. For example, when the annotator is $90\%$ sure of the labeling, we flip the labels of the transition segments with probability $0.1$. We decrease the confidence level by $10\%$ at each time to verify the impact of labeling uncertainty.

In practice, the confidence values of labeling are hard to obtain for a single annotator. For example, it is quite clear for the annotator that which two labels should be associated with the transition segment, however the annotator cannot give a specific confidence values for that. Therefore, when evaluating our soft labeling method, we assume two possible labels of the transition segment are equally likely to be assigned, *i.e.* we assign soft labeling of the transition segment as

$$\pi_k(y) = \begin{cases} 0.5 & \text{if } y = y_{k-1} \text{ or } y = y_{k+1} \\ 0 & \text{otherwise} \end{cases}$$

where $\pi_k$ is the soft labeling vector for segment $k$, and $\pi_k(y)$ refers to the confidence of assigning the segment to label $y$. $y_{k-1}$ is the label from the previous segment and $y_{k+1}$ is the label of the next segment.

We compare our approach against four baseline methods. The first baseline uses a linear-chain CRF for activity recognition [19]. The second baseline is adopted from [10], which assumes multiple hypotheses of video segmentation. We report both the average and the best results over different segmentation methods. The third baseline [10] uses majority voting to assign the labels, *i.e.* the label that is predicted by most of the segmentations is chosen. The fourth baseline is Hu et al. [5]. As has been discussed in Section II, this work adopts latent variables to model human activities and is shown to outperform the work of Koppula et al. [10]. We therefore consider it as the state-of-the-art approach.

The models are trained using the Structured-SVM framework [16]. For optimization we use the 1-slack algorithm (dual) described in [7]. All results are evaluated over number of latent states ($L$) from 1 to 10. The optimal number of latent states and the SVM parameters are chosen by cross-validation.

## B. Data

The CAD-120 dataset contains over 120 RGB-D videos with 4 subjects performing daily life activities, and it contains a total number of 61,585 RGB-D video frames. There are in total 10 different sub-activities to be recognized, *i.e.* reaching, moving, pouring, eating, drinking, opening, placing, closing, scrubbing, and null. Start and end frame of these activities are labeled.

The videos are discretized into temporal segments. Following [2, 10], we apply a graph-based approach to parse the video into temporal segments. Specifically, we use a node to represent the skeleton joints at each frame. Edge features connecting the nodes are measured as the replacement and speed of the joints between two frames. Such a model favors stable changes of the skeleton joints, and videos are more likely to be split when the change is large. The segmentation method starts with each frame as a single segment, and then it merges similar frames iteratively. We use the same set of parameters for video segmentation as described in [5, 10]. In addition, we also use a uniform segmentation for comparison. The results are averaged over the segmentation with different parameters. We choose the CAD-120 dataset for evaluation because of the following reasons: 1) CAD-120 is a very challenging dataset that presents significant variations of activities, cluttered background, viewpoint changes, and partial occlusions. 2) The dataset has been used in many recent works in the robotics research [5, 10]. Therefore we can easily compare the performance to the state-of-the-art approaches. 3) The dataset is captured by a RGB-D camera mounted on the robot, which is closely related to the applications in robotics. For comparison, the same input features are used as in [10].

## C. Evaluation

All results reported in this paper are evaluated with 4-fold cross-validation. The folds are split based on different subjects, *i.e.* the model is trained on videos of 3 persons and tested on a new person. Each cross-validation is run for 3 times. To check the generalization of our model across different data, the results are averaged across the folds.

In this paper, accuracy (classification rate), precision, recall and F1-score on the test data are reported for comparing different models. In the CAD-120 dataset, more than half the instances of the dataset are “reaching” and “moving”.

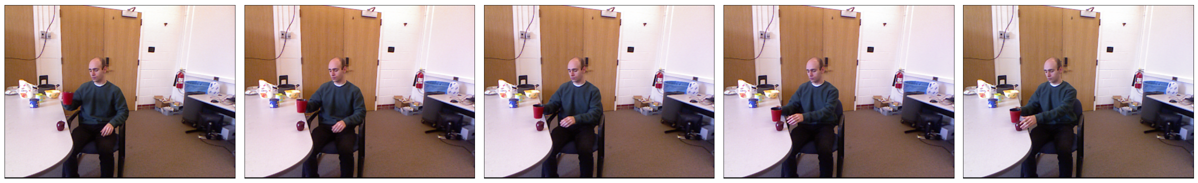


Fig. 5. A transition segment where the label is "wrongly" classified by the soft labeling model. The "groundtruth" label provided by the CAD120 dataset is *reaching*, and the prediction of our model is *moving*. We can see that the person is performing two activities at the same time, *reaching* for an apple (left hand) and *moving* a cup (right hand). Assigning a single label to these frames is problematic. In contrast, the soft labeling method is able to capture both of the activities and the inappropriate annotation is corrected by learning with soft labeling.

Therefore we consider precision, recall and F1-score being relatively better evaluation criteria than accuracy, as they remain meaningful despite class imbalance.

## D. Latent Variables Initialization

The latent variables (*i.e.* $\mathbf{z}$ and $\mathbf{h}$) need to be initialized before training. The hidden variables $\mathbf{z}$ are initialized using a data-driven approach. We apply K-means clustering on the input data $\mathbf{x}$, and we assign the cluster labels as the initial latent states. To reduce the risks of converging to local minima, we repeat the K-means algorithm for 10 times. Then we choose the best clustering results that have the minimal within-cluster distances.

We applied a pre-learning approach to initialize the latent activity labels $\mathbf{h}$. We split the training data at the position of uncertain labels to form a new dataset where the labels are complete. After the model parameters are learned, we can initialize the uncertain activity labels using Eq. (5).

## E. Results

Fig. 4 compares the performance of our method with two baseline models at different annotation confidence levels. Confidence level $100\%$ indicates that the models are trained based on the ground truth labels and no label is changed. Conversely, if the annotator is completely ambivalent about the labels at all transition segments, *i.e.* the confidence level is $0\%$, we simulate the annotation by changing the labels for all transition segments.

*1) How detrimental is hard labeling:* Fig. 4 shows the performance of two baseline approaches ([5] in dashed lines and the state-of-the-art approach [19] in dotted lines). We can see that both of the baselines drop dramatically with the decreasing confidence level. Both of the baselines can only learn from data that are labeled with hard assignment. When the data is complex, the human annotators become less confident of the labels and they are more likely to make mistakes. Hard labeling does not incorporate the uncertainty of data labeling, but rather the erroneous labels are treated as the true ones for learning the model parameters. The drop of the performance suggests that hard labeling is very sensitive to mistakes, and it is therefore risky to use hard labeling on complex data. Fig. 5 illustrates a transition segment that is "wrongly" classified by the soft labeling method. In this example, the person is performing two activities at the same time. Hard labeling to these frames is very harmful because only one label can be preserved. Using soft labeling, both activity labels can be considered in learning, and the most suitable label of the segment is predicted.

*2) How much can we benefit from using soft labeling:* Instead of using hard assignment, we use soft labeling in our model. The performance of soft labeling (solid lines in Fig. 4) drops gradually with a slower rate compared to the decrease rate of the confidence. Notably, we show that our model retains almost the same performance as using the ground truth data until the confidence drops below $60\%$. When the confidence is $0\%$, we have a gain of over 10 percentage points in both precision and recall compared with the baseline approaches. Interestingly, with a small part of the labels being changed, at the confidence of $90\%$, soft labeling with the uniform segmentation achieves better results than using the "ground truth" labels from the CAD-120 dataset (confidence of $100\%$). This suggests a small part of the "ground truth" labels are incorrect, and our model can recapture the true labels by using soft labels.

Table II compares the F1-score of the approaches quantitatively. As annotators are less confident, the benefit of soft labeling becomes more obvious, outperforming the baselines by $10\%$ when the confidence of transition labels is $0\%$. Results of the T-test show that our model performs significantly better ($p < 0.05$) than the state-of-the-art approach at confidence levels from $10\%$ to $90\%$. Table II also compares different approaches that use a single segmentation (*Avg. segmentation* and *Best segmentation* [10]), multiple segmentation methods (*Majority voting* [10]), and soft labeling. The best single segmentation performance outperforms the average by around 3 percentage points. In practice, however, it is hard to decide which single segmentation method is the best to use because the best segmentation method on the training data may not generalize well to new data. *Majority voting* combines all segmentation hypotheses, and it outperforms the *Best segmentation* method in most of the cases. However, *Majority voting* is very expensive in both training and testing compared with soft labeling, and the execution time grows with the number of segmentation hypotheses. Besides, *Majority voting* does not incorporate any labeling confidence during training. In contrast, our model builds upon a unified framework which incorporates confidence of labels for training the model, which explains its better performance.

*3) Effect of latent variables:* We investigated the role of the latent variables and visualize how they relate to our intuition.

TABLE II
PERFORMANCE OF F1-SCORE ON THE CAD-120 DATASET. STANDARD ERROR IS ALSO REPORTED.

| Graph-based segmentation | | | | | | | |
|---|---|---|---|---|---|---|---|
| Confidence Level | 100% | 90% | 80% | 60% | 40% | 20% | 0% |
| Linear CRF [19] | 68.4±1.0 | 66.9±1.4 | 66.0±1.1 | 63.3±1.4 | 60.4±1.5 | 56.8±1.7 | 51.9±1.7 |
| Avg. segmentation [10] | 63.7±2.3 | 62.7±2.0 | 62.9±2.1 | 61.7±1.8 | 59.9±2.4 | 57.1±1.8 | 54.2±2.0 |
| Best segmentation [10] | 65.5±1.5 | 64.6±1.4 | 63.7±1.5 | 63.3±2.2 | 61.2±2.5 | 59.4±1.5 | 56.6±2.7 |
| Majority voting [10] | 65.2±2.9 | 63.5±2.9 | 64.3±2.9 | 63.0±1.8 | 62.0±2.2 | 59.5±1.3 | 56.7±2.1 |
| Hu et al. [5] | 69.0±1.2 | 68.1±1.2 | 67.0±1.2 | 64.3±1.4 | 61.2±1.4 | 57.9±1.7 | 53.3±1.5 |
| Our method | **69.0±1.2** | **68.7±1.5** | **68.8±1.4** | **68.3±1.1** | **66.6±1.3** | **64.7±1.3** | **62.6±1.5** |
| Uniform segmentation | | | | | | | |
| Linear CRF [19] | 67.8±1.3 | 67.1±1.3 | 65.9±1.2 | 62.9±1.8 | 58.2±2.0 | 54.6±2.1 | 50.6±1.5 |
| Avg. segmentation [10] | 60.3±2.8 | 60.9±2.8 | 60.3±2.6 | 59.1±2.4 | 57.6±2.8 | 56.7±2.6 | 53.1±2.8 |
| Best segmentation [10] | 63.1±2.1 | 64.3±2.3 | 63.7±1.7 | 63.4±2.3 | 60.6±2.8 | 59.6±2.8 | 56.4±2.3 |
| Majority voting [10] | 64.3±3.0 | 64.0±3.1 | 64.2±2.6 | 64.1±2.3 | 61.9±3.0 | 60.6±2.7 | 56.0±3.0 |
| Hu et al. [5] | 68.9±1.5 | 68.2±1.5 | 66.7±1.6 | 63.9±2.1 | 59.1±2.2 | 55.2±2.5 | 51.1±2.5 |
| Our method | **68.9±1.1** | **69.1±1.5** | **68.2±1.5** | **67.1±1.6** | **65.9±1.8** | **64.1±1.4** | **61.2±2.0** |

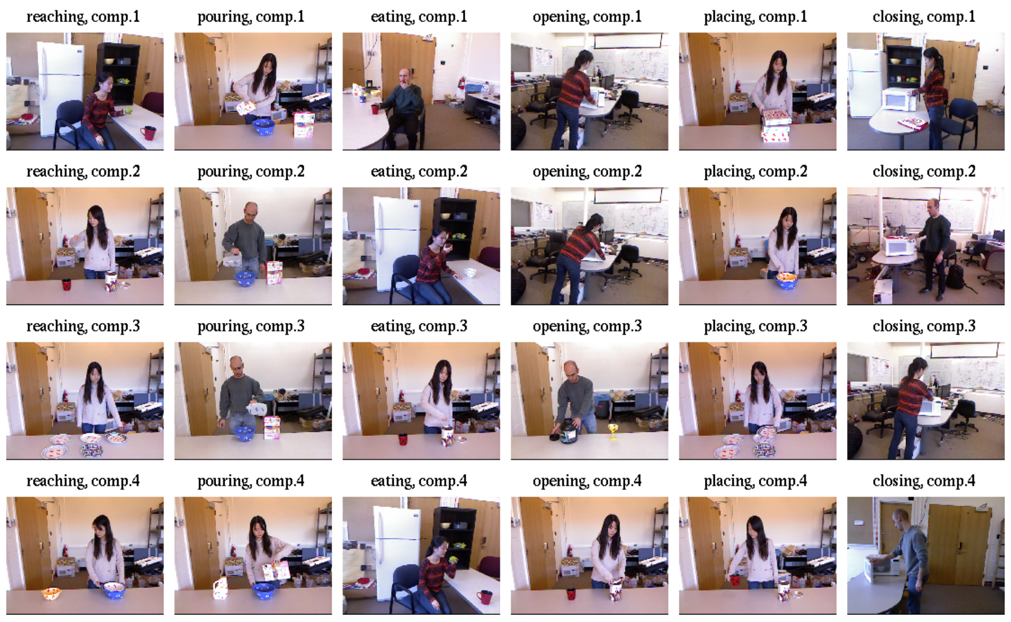


Fig. 6. Visualization of the latent components. The columns are six activities and rows refer to the four latent components. Due to the limitation of space, here only 6 activities are illustrated. See more examples at `http://ninghanghu.eu/index.html#rss14`

We visualize the 4 latent states of our model by sampling frames for each activity and latent component pair. Fig. 6 shows that the activities vary largely over different actors and the objects that the actors are manipulating with. *e.g.* For the activity of *eating*, the viewpoint can be front (comp. 1) or side view (comp. 3). The actor can be right-handed (comp. 2) or left-handed (comp. 4). For the activity of *opening*, we can see that the latent components capture the difference between opening a door (comp. 1 and 2) and opening a lid (comp. 3 and 4). This illustrates the large range of variation that can be found in humans activities, and how our model tries to capture as much as possible of this variation in the limited cardinality of the latent variables. These latent variables are able to capture the sub-level semantics that are informative for activity recognition.

## VII. CONCLUSION

In this paper, we present a method to train discriminative graphical models, which allows annotation uncertainty to be explicitly incorporated, in the form of soft labeling. The advantage of soft labeling is that it incorporates the uncertainty of labels during annotation and can deal with missing labels or annotator disagreement. As soft labeling is a very general representation, our model can be extended to solve other tasks, *e.g.* learning with partially annotated data, unsupervised learning. We propose a novel approach for learning, where computation of the loss functions considers the annotation uncertainty. We evaluate the framework on the benchmark dataset for activity recognition. Our cross-validated results on the CAD-120 dataset demonstrate the benefits of soft labeling in handling annotation uncertainty at transition.

## Acknowledgments

The research is funded by the European project ACCOMPANY (grant agreement No. 287624) and the European project MONARCH (grant agreement No. 601033).

We thank Joris Mooij for useful discussions and we also thank Hema Koppula for providing us with the CAD-120 dataset and details of her implementation.

## References

[1] F. Amirabdollahian, S. Bedaf, R. Bormann, and Others. Assistive technology design and development for acceptable robotics companions for ageing years. *Paladyn, Journal of Behavioral Robotics*, pages 1–19, 2013.

[2] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision (IJCV)*, 59(2):167–181, 2004.

[3] N. Hu, G. Englebienne, and B. Kröse. Posture Recognition with a Top-view Camera. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2152–2157, 2013.

[4] N. Hu, G. Englebienne, and B. Kröse. A Two-layered Approach to Recognize High-level Human Activities. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (ROMAN)*. IEEE, 2014.

[5] N. Hu, G. Englebienne, Z. Lou, and B. Kröse. Learning Latent Structure for Activity Recognition. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[6] Y. Jiang and A. Saxena. Infinite Latent Conditional Random Fields for Modeling Environments through Humans. In *Proceedings of the Robotics Science and Systems (RSS)*, 2013.

[7] T. Joachims, T. Finley, and C. N. Yu. Cutting-plane training of structural SVMs. *Machine Learning*, 77(1): 27–59, 2009.

[8] H. Koppula and A. Saxena. Learning Spatio-Temporal Structure from RGB-D Videos for Human Activity Detection and Anticipation. *Proceedings of the International Conference on Machine Learning (ICML)*, 2013.

[9] H. S. Koppula and A. Saxena. Anticipating human activities using object affordances for reactive robotic response. In *Proceedings of the Robotics Science and Systems (RSS)*. RSS, 2013.

[10] H. S. Koppula, R. Gupta, and A. Saxena. Learning Human Activities and Object Affordances from RGB-D Videos. *International Journal of Robotics Research (IJRR)*, 32(8):951–970, 2013.

[11] X. Lou and F. Hamprecht. Structured learning from partial annotations. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1519–1526, 2012.

[12] J. M. Mooij. libDAI: A Free and Open Source C++ Library for Discrete Approximate Inference in Graphical Models. *Journal of Machine Learning Research*, 11: 2169–2173, Aug. 2010.

[13] U. Reiser, C. Connette, J. Fischer, J. Kubacki, A. Bubeck, F. Weisshardt, T. Jacobs, C. Parlitz, M. Hagele, and A. Verl. Care-O-bot 3-Creating a product vision for service robot applications by integrating design and technology. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1992–1998. IEEE, 2009.

[14] J. Sung, C. Ponce, B. Selman, and A. Saxena. Unstructured human activity detection from rgbd images. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 842–849. IEEE, 2012.

[15] K. Tang, L. Fei-Fei, and D. Koller. Learning Latent Temporal Structure for Complex Event Detection. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1250–1257. IEEE, 2012.

[16] I. Tsochantaridis and T. Hofmann. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the International Conference on Machine Learning (ICML)*, page 104. ACM, 2004.

[17] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, pages 1453–1484, 2005.

[18] A. Vahdat and G. Mori. Handling uncertain tags in visual recognition. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2013.

[19] D. L. Vail, M. M. Veloso, and J. D. Lafferty. Conditional Random Fields for Activity Recognition. In *Proceedings of the international joint conference on Autonomous agents and multiagent systems (AAMAS)*, page 235. ACM, 2007.

[20] T. Van Kasteren, A. Noulas, G. Englebienne, and B. Kröse. Accurate activity recognition in a home setting. In *Proceedings of the International Conference on Ubiquitous Computing*, pages 1–9. ACM, 2008.

[21] S. B. Wang and A. Quattoni. Hidden conditional random fields for gesture recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1521–1527. IEEE, 2006.

[22] Y. Wang and G. Mori. Max-margin hidden conditional random fields for human action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 872–879. IEEE, 2009.

[23] D. Wyatt, M. Philipose, and T. Choudhury. Unsupervised Activity Recognition Using Automatically Mined Common Sense. In *Proceedings of the National Conference on Artificial Intelligence*, pages 21–27, 2005.

[24] C. N. Yu and T. Joachims. Learning structural SVMs with latent variables. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 1169–1176. ACM, 2009.

[25] A. Yuille and A. Rangarajan. The concave-convex procedure (CCCP). *Advances in neural information processing systems (NIPS)*, 2:1033–1040, 2002.