Learning Human Activities and Object Functionality through Conditional Random Fields

Nakul Gopalan
Department of Computer Science
Brown University
Providence, RI 02912
ngopalan@cs.brown.edu

Jun Ki Lee
Department of Computer Science
Brown University
Providence, RI 02912
jklee@cs.brown.edu

Abstract

Recognizing a human activity is a skill that humans learn early in life. Activity recognition helps humans in predicting next steps of the activity in question and also in planning other activities with respect to the recognized activity. Our goal is to recognize human activities using machine learning, to predict next steps of an action, or provide help using robots or other interfaces. Specifically, what interests us is the human activities being performed near a table-top table or a shelf, where the human can be observed and helped by a robot. We will use a dataset from [2] to learn a Graphical Model with temporal and spatial structure, which can then be used to predict the action being performed and functionalities of objects. Both can then be used to recognize a higher level activity. First we compare the graph structures used in the original paper to understand the significance of the graph structure used. Further, our efforts in this work are to model the problem as an HMM first and then a CRF (Conditional Random Fields) and compare performance to the original paper in classifying sub-activities. We classify the main activity using a bag of words model with features from the complete activity.

1 Introduction

Recognizing a human activity is a skill that humans learn early in life. Activity recognition helps humans in predicting next steps of the activity in question and also in planning other activities with respect to the recognized activity. Our objective is to recognize human activities using machine learning, to predict next steps of an action. Recognized actions can provide help to humans using robots or other interfaces. Specifically of interest to us are the human activities being performed on a table-top, where the human can be observed and helped by a robot.

We use the dataset[1] from [2] to learn a Graphical Model with temporal and spatial structure, which can then be used to predict the action being performed and functionalities of objects. The dataset is created from streams of the RGB-D images acquired from a Infrared depth sensing camera. The dataset contains both the tracked human skeletal poses and object locations and supervised labels of sub- and high-level- activities and object functionalities. We first provide some background over the methods used to solve this problem before and an overview of the dataset. Next we alter the original graph structure in [2] by changing their code and running learning and testing on the new graph structure. Finally we try to predict the sub-activities using linear chain HMMs and CRFs.

[1] http://pr.cs.cornell.edu/humanactitivities

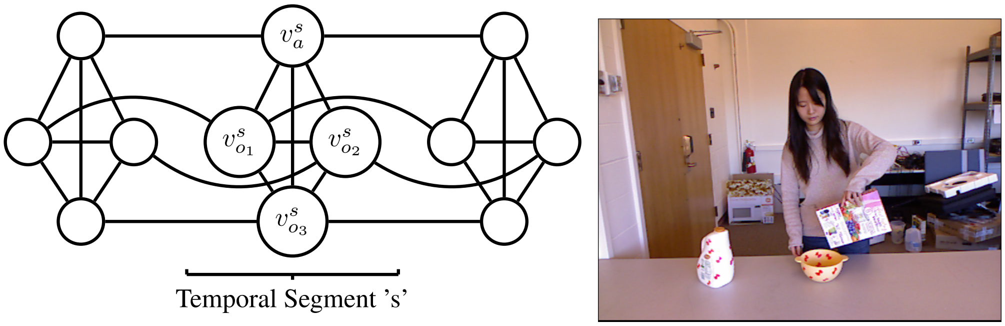


Figure 1: **(a)** an example of a Conditional Random Field used in [2]. Each temporal segment (s) has one sub-activity node v_a^s and three object nodes $\{v_{o_1}^s, v_{o_2}^s, v_{o_3}^s\}$, **(b)** an example image from the data set

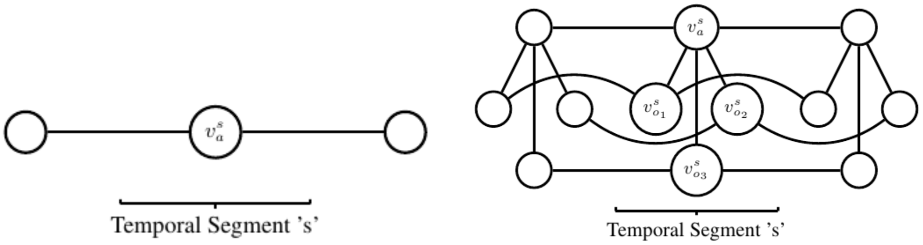
2 Previous work

3 Dataset

We explored data sets with activity labels and corresponding skeletal trajectories. Humans frequently perform activities using objects, especially in table-top scenarios. The dataset from [2] was constructed keeping this relationship in mind. The raw data consists of the RGB-D frames from the scenes of humans performing activities and human skeletal poses from the OpenNI NITE tracker [4]. The dataset has recordings of 120 labelled moderate sized human activities of ten types, e.g.: making cereal, microwaving, taking food, etc. The skeletal poses are not precise and have confidence intervals. The labels on the data are related to both the objects present in the scene and the activities involved. Activity specific labels are frame-wise labels on sub-activities and the main activity being performed. The sub-activity labels are that of *reaching, moving, pouring, eating, drinking, opening, placing, closing, scrubbing* and *null*. The object specific data consist of object locations for each time frame per object with its object id and its affordance present in the scene. The object affordance labels are those of *reachable, movable, pourable, pour to, containable, drinkable, openable, placeable, closable, scrubbable* and *stationary*. The dataset also has pre-computed features that have relationship between objects and skeletal poses across frames.

4 Methods

The data from [2] has both temporal and spatial structure. For example, the spatial structure has relationships between objects and skeleton positions within a frame. There is also a temporal relationship between previous skeletal pose and object positions and those of the current frame. The authors from [2] modeled this structure using a pairwise Conditional Random Field([7]) as shown in Fig. **??**. Here each node corresponds to an object or a sub-activity. The object nodes can take any of the 12 possible affordance labels and the sub-activity nodes can take any of the 10 possible labels. The affordance and subactivity nodes are hidden, but feature values for label assignments over the nodes can be computed. The authors in [2] first segmented an input activity, or set of frames into segments that are more likely to belong to a single sub-activity. The authors used a sum of Euclidean distance between joints metric to judge the change in sub-activities. The threshold to split sub-activities seems to be based on a trial and error heuristic. The sub-activities are temporally linked giving a chain graph structure as shown in Fig. **??**. The authors used this graph structure to predict sub-activity labels and object affordances by performing inference over the model, and then used an SVM to predict the label of the over all activity. To predict the sub-activity and object affordance labels the authors used a maximum energy formulation, where the labels that lead to the maximum possible energy together are accepted. This energy is a sum of products of features and feature weights. The energy function requires learning the weights of the features first, with the supervised labeled input from the dataset discussed above, such that training error is minimized.



(a) Graph with only sub-activity labels

(b) Graph with no object - object edges

Figure 2: Different graph structures tried

5 Importance of graph structures

The original paper by [2] uses the graph structure shown in Fig. **??**. The sub-activity nodes and object affordance nodes are completely connected within a temporal segment. The authors infer the series of sub-activities as an argmax over an energy function for the linear chain that describes the sequence of sub-activities:

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}}\, E_{\mathbf{w}}(\mathbf{x}, \mathbf{y}) \tag{1}$$

$$E_{\mathbf{w}}(\mathbf{x}, \mathbf{y}) = E_o + E_a + E_{oo} + E_{oa} + E_{oo}^t + E_{oa}^t \tag{2}$$

$$E_o = \sum_{i \in \mathcal{V}_o} \psi_o(i) = \sum_{i \in \mathcal{V}_o} \sum_{k \in K_o} y_i^k \left[\mathbf{w}_{\mathbf{o}}^k \cdot \phi_o(i)\right] \tag{3}$$

$$E_a = \sum_{i \in \mathcal{V}_a} \psi_a(i) = \sum_{i \in \mathcal{V}_a} \sum_{k \in K_a} y_i^k \left[\mathbf{w}_{\mathbf{a}}^k \cdot \phi_a(i)\right] \tag{4}$$

$$E_{oo} = \sum_{(i,j) \in \mathcal{E}_{oo}} \sum_{(l,k) \in K_o \times K_o} y_i^l y_j^k \left[\mathbf{w}_{\mathbf{oo}}^{lk} \cdot \phi_{oo}(i,j)\right] \tag{5}$$

$$E_{oa} = \sum_{(i,j) \in \mathcal{E}_{oa}} \sum_{(l,k) \in K_o \times K_a} y_i^l y_j^k \left[\mathbf{w}_{\mathbf{oa}}^{lk} \cdot \phi_{oa}(i,j)\right] \tag{6}$$

$$E_{oo}^t = \sum_{(i,j) \in \mathcal{E}_{oo}^t} \sum_{(l,k) \in K_o \times K_o} y_i^l y_j^k \left[\mathbf{w}_{\mathbf{oo}}^{t^{lk}} \cdot \phi_{oo}^t(i,j)\right] \tag{7}$$

$$E_{aa}^t = \sum_{(i,j) \in \mathcal{E}_{aa}^t} \sum_{(l,k) \in K_a \times K_a} y_i^l y_j^k \left[\mathbf{w}_{\mathbf{aa}}^{t^{lk}} \cdot \phi_{aa}^t(i,j)\right] \tag{8}$$

where $E_w(\mathbf{x}, \mathbf{y})$ is the total energy over all the temporal segments in the activity, $\mathbf{x}$ are the features and $\mathbf{y}$ are the sub-activity and object affordance labels. $\hat{\mathbf{y}}$ is the inferred series of object affordance and sub-activity labels. V_a and V_o are the sub-activity and object nodes of the graph. K_a and K_o are the sub-activity and object affordance labels that the nodes can take. E_o and E_a are the energy functions over individual nodes, i.e., sums over all the nodes taking on all possible combinations of the labels y^k with given features ϕ, when the weights of the nodes for label k are $\mathbf{w}^k$. Further there are energy terms over the edges of the graph E_{oo} and E_{oa} that sum over all possible combination of labels between object - object nodes or object - sub-activity nodes at each temporal segment, given features and weights. These represent the relationship between object affordances and object affordance and sub-activities. There are also temporal energy terms of E_{oo}^t and E_{aa}^t that represent the temporal relationship between different sub-activity labels or different object affordance labels, w.r.t. the labels and features of the temporal segment.

During learning the weights $\mathbf{w}$ are learned such that the error between the predicted label $\hat{\mathbf{y}}$ and true label $\mathbf{y}$ is minimized. Koppula et al. [2] formulate this problem like a Structural Support Vector Machine problem, which is a convex quadratic problem, and solve it using the cutting plane algorithm. Further after learning the weights, inference is performed using the learned weights to

Table 1: Comparison of Results with different graph structures.

Graph structure	Object Affordances		Sub-activity Labels	
	Precision	Recall	Precision	Recall
Sub-activity Only	NA	NA	76%	76%
Sub Activity and Object Affordances (No Obj-Obj Relationships)	87%	87%	88%	88%
Full Model	92%	92%	89%	89%

predict the most likely sequence of sub-activities. This inference is performed using mixed integer programming.

We tried to understand the importance of the edges in the graph structure in predicting sub-activity labels. First we performed learning on the original graph Fig. **??** with the author's source code. The sub-activity labels were predicted correctly 89% of the times as shown in the Table 1. Next we removed all dependence of the graph on the object affordance labels, by tying the weights related to the object affordance nodes to zero, and not performing learning over these weights. We get a linear chain graph in this case, Fig 2a, and the precision results were 76% clearly lower than those with the object-affordance labels. Further we added back edge weights between sub-activities and object nodes but not between object - object nodes: Fig 2b. We noticed that this lead to a large increase in the precision number to 88%. Hence the object-object edges within the same temporal segment in the graph are useful for about 1% increase in precision.

In this section we looked at the importance of graph structure in solving this inference problem. Next we look at the importance of temporal segmentation in learning and inference.

6 Modeling without temporal segments using a simple HMM

The idea for this experiment was that the simplest way to formulate this problem would be an HMM over all the frames of the data, and predicting the hidden state, i.e., the sub-activity label per frame instead of predicting them for a temporal segment. We performed a training-test split of $2 : 1$ and learned transition probability between states and observation probabilities of skeletal features. Since the skeletal features were sparse we reduced them to low 10 dimensional features using clustering. What we noticed immediately was that the transition matrix was almost an identity matrix as there are very few transitions between frames from one sub-activity to another, and far higher self transitions. This lead to the output predictions of sub-activity for all frames being stuck to the most likely sub-activity, producing a low result of 16% accuracy where 10% is chance. We next try to train a CRF based model for temporally segmented data.

7 Naïve Activity Detection

We classify high level activities from skeletal trajectories using a Bag of Words model [5] with skeletal poses as features. For each type of activity we have about 12 examples in the dataset. We use about 8 examples per activity for training and 4 examples per activity for testing. Each frame has a skeletal pose, of about 170 dimensions, and each activity has about 400 frames. We cluster skeletal poses over all training activities into 200 clusters using k-means. Further for each activity label we create a histogram by counting number of skeleton frames of an activity in each of the 200 clusters. Next for each test activity we find the nearest neighbor word from the trained histogram words and return its label. The accuracy of this simple Bag of Words model is about 67.3%. The original paper cites an accuracy of 75% in recognizing these high level activities, but when we ran experiments with their current code base this number was higher than 90%. The confusion matrix for this classification is show in Fig 3. It can clearly been seen that there is confusion in classification of the activity labels of stacking and unstacking as differentiating between these labels needs temporal information. Similar confusion was also seen in the original paper.

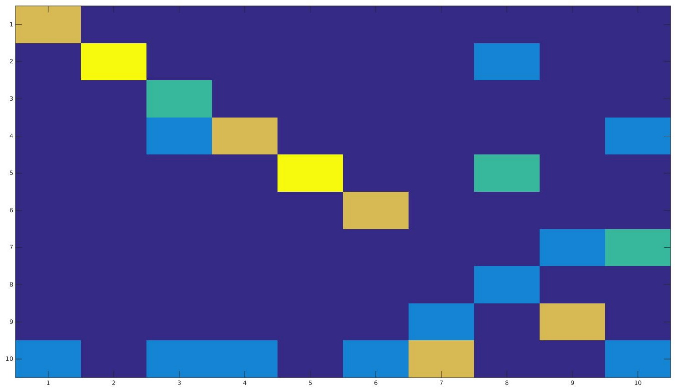


Figure 3: Confusion Matrix of the high level activity classification using Bag of Words model. The label order is - arranging objects, cleaning, having meal, making cereal, microwaving food, picking objects, stacking, taking food, taking medicine, unstacking objects

8 Conclusion

In this work we tried different methods to learn the hidden labels of sub-activities given the dataset from [2]. We saw that changing the graph structure w.r.t. to the original formulation leads to a drop in performance. The temporal segmentation of frames is a key requirement in this framework. Further that naïve methods like bag of words models can go a long way in guessing high level activity labels.

References

[1] Juergen Gall, Andrea Fossati, and Luc Van Gool. Functional categorization of objects using real-time markerless motion capture. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1969–1976. IEEE, 2011.

[2] Hema Koppula, Rudhir Gupta, and Ashutosh Saxena. Learning human activities and object affordances from rgb-d videos. *IJRR*, 2013.

[3] Hema Swetha Koppula and Ashutosh Saxena. Anticipating human activities using object affordances for reactive robotic response. In *Robotics: Science and Systems*, 2013.

[4] PrimeSense Inc. *Prime Sensor NITE 1.3 Algorithms notes*, 2010.

[5] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*, 2003.

[6] Jaeyong Sung, Colin Ponce, Bart Selman, and Ashutosh Saxena. Unstructured human activity detection from rgbd images. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 842–849. IEEE, 2012.

[7] Charles Sutton and Andrew Mccallum. *Introduction to Conditional Random Fields for Relational Learning*. MIT Press, 2006.