Learning Action Symbols for Hierarchical Grammar Induction

Kyuhwa Lee, Tae-Kyun Kim and Yiannis Demiris
Department of Electrical and Electronic Engineering, Imperial College London
{*k.lee09, tk.kim, y.demiris*}*@imperial.ac.uk*

Abstract

We present an unsupervised method of learning action symbols from video data, which self-tunes the number of symbols to effectively build hierarchical activity grammars. A video stream is given as a sequence of unlabeled segments. Similar segments are incrementally grouped to form a hierarchical tree structure. The tree is cut into clusters where each cluster is used to train an action symbol. Our goal is to find a good set of clusters i.e. symbols where regularities are best captured in the learned representation, i.e. induced grammar. Our method has two-folds: 1) Create a candidate set of symbols from initial clusters, 2) Build an activity grammar and measure model complexity and likelihood to assess the quality of the candidate set of symbols. We propose a balanced model comparison method which avoids the problem commonly found in model complexity computations where one measurement term dominates the other. Our experiments on the towers of Hanoi and human dancing videos show that our method can discover the optimal number of action symbols effectively.

1 Introduction

Representation and recognition of human activities is an important area of computer vision where an activity is composed of multiple atomic action components. In syntactic approaches, these atomic action components are represented as symbols similar to the concept of vocabularies in languages, which could form a hierarchical structure e.g. $(a^n b^n)^m$ (n and m repetitions of two action symbols a and b), depending on the frequency of action symbols occurring [1, 11, 2].

Activities are often represented and inferred by context-free grammar (CFG) or Stochastic context-free grammar (SCFG) techniques, owing to their expressiveness and robustness to noise. In [3], Ivanov uses SCFGs to recognize complex actions, e.g. music conducting gestures, using HMM-based action symbols. In [10], Ota et al. use SCFGs to describe the structures of Kanji using few stroke shapes and relative position labels. As opposed to manually defining the grammar rules [3, 10], there are several works aimed at constructing (or inducing) grammars from repetitive action symbols [12, 13, 4]. Solan et al.[12] induced CFGs using graphical representations, subject to no recursions. Stolcke and Omohundro [13] presented a SCFG induction technique, and more recently it has been extended by Kitani et al. [4] and Lee et al. [7] to cope with noisy symbols and symbol uncertainties respectively. However, the aforementioned studies are limited in the sense that action symbols are predefined.

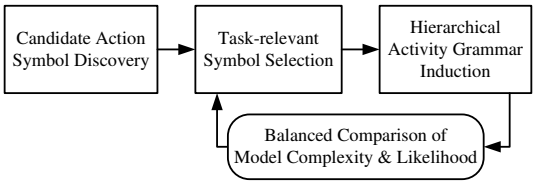

Figure 1. Overview: Candidate symbols are generated using agglomerative hierarchical clustering approach, where too general or specific symbols are subsequently filtered out by measuring the model complexity and likelihood.

In this work, we propose a system which can discover a meaningful set of symbols and represent activities using these symbols. We consider the towers of Hanoi and human dancing videos as input, where certain symbols appear regularly. As we define more symbols, the description complexity increases, resulting in overfitting as it captures all the subtle differences in human movements. On the other hand, if we define too small number of symbols, it cannot capture the meaningful hierarchies of actions, resulting in underfitting. We are interested in finding the balancing point using the minimum description length of the induced grammar. In our approach, we first discover a number of candidate models where each model has a different set of symbols. Next, we induce stochastic hierarchical grammars in an unsupervised way using a method described in [7] for every

candidate model, which feedbacks a model description length and likelihood value that are used to select models (Fig.1). Since the value ranges of prior probability and likelihood differ in large amount, we propose a balanced (non-dominating) comparison between these two measurements using Pareto optimality to assess the qualities of the chosen symbols.

Kulic et al.[5] proposed a method to incrementally add observed actions in a hierarchical tree structure, where leaf nodes represent specific motions and more generalized nodes are located closer to the root. The tree is cut into clusters where each cluster corresponds to each symbol. Our method is distinguished by how we measure the validity of the learned activity grammars to choose a better set of symbols (i.e. the feedback). Liang et al.[8] trained variable-length Markov models (VLMM) which can automatically learn the model parameters of atomic human actions, where each VLMM is trained to learn each unlabeled action symbol. In our case, however, we do not assume how many action symbols are available. Whereas the codebook and topic models are sequentially obtained for learning action categories in [9, 14], action symbols and activity grammars are found with the feedback in this paper.

2 Approach

As we are concerned with learning syntactic-level action symbols, we preprocess input video sequences into a series of vector representations using low-level feature descriptors. The choice of a low-level descriptor depends on the problem domain, e.g. joint-space description for human motion capture data. Each vector is defined as a group of consecutive frames which share the similar low-level descriptions within the group. They can be regarded as unlabeled video segments.

2.1 Discovery of Candidate Symbols

We begin our method by clustering segment vectors using hierarchical agglomerative clustering which incrementally builds a binary tree by grouping a pair of similar vectors based on some distance function, starting from leaf nodes (single-vector nodes). The height of a node represents a distance between two child nodes. By grouping nodes with height less than τ, we obtain κ clusters of vectors. We set initial τ_κ

$$\tau_\kappa = max(\chi(i,j)) \quad \forall i,j \tag{1}$$

where inconsistency coefficient $\chi(i,j)$ measures how objects contained in child nodes i and j differ from each other:

$$\chi(i,j) = \frac{d(i,j) - \mu_{i,j}}{\sigma_{i,j}} \tag{2}$$

with $\mu_{i,j}$ and $\sigma_{i,j}$ respectively representing mean and standard deviation of heights of all subnodes of i and j.

$$d(i,j) = \sqrt{\frac{2n_i n_j}{n_i + n_j}} \|\bar{x}_i - \bar{x}_j\|_2 \tag{3}$$

is a distance function defined using Ward's method to take into account the cost of merging two clusters. Intuitively, the higher the value of $\chi(i,j)$, the less similar the objects belong to that link, hence inconsistent.

The mean of each cluster is used as a symbol description that can classify input video segments and label with its symbol index. We represent a system having κ symbols as ψ_κ. However, as we do not have prior knowledge about whether using κ symbols is optimal to represent an activity effectively, different number of symbols need to be tested: $\Psi = \{\psi_1, \psi_2, ..., \psi_\kappa\}$.

An advantage of using hierarchical clustering analysis is that it does not depend on initial conditions unlike k-means and provide an intuitive way to partition data points into a desired number of clusters.

2.2 Selecting the Number of Symbols

For each system $\psi_\kappa \in \Psi$ obtained in the last section, we build an activity representation from data using acquired symbols. We require that our training method is able to 1) obtain model parameters in unsupervised way, 2) measure model complexity and likelihood at any stage of training, and 3) deal with recursions.

We choose stochastic context-free grammars (SCFGs) as our underlying representation and adopt SCFG induction technique because such a framework provides a compact way to represent hierarchical and recursive structures, and their unsupervised learning algorithms rely on MDL principle which is used as feedback score in our case.

2.2.1 Computation of MDL scores

As described in [6, 7], a SCFG is learnt from data in an unsupervised way by iteratively applying two types of operators, *Substitute* and *Merge*, until the best grammar is found based on MDL principle. The objective is to find a representation that is sufficiently simple yet expressive. It is reported in [7] that lower MDL scores generally lead to a better representation, based on real-world experiments. By measuring prior probability of a model $P(M)$ and data likelihood $P(D|M)$, our goal is to minimize the MDL score, represented as $-log$ of joint probability $P(M,D)$:

$$-logP(M,D) = -logP(M) - logP(D|M) \tag{4}$$

$$P(M) = P(M_S, M_\theta) = P(M_S)P(M_\theta|M_S) \tag{5}$$

where $P(M_S)$ denotes structure prior and $P(M_\theta)$ denotes parameter prior. Both are defined in the same way as in [7, 4, 13]. $P(D|M)$ term is typically computed using Viterbi parsing, but to take into account the uncertainty values of the input symbols, we use the SCFG parsing algorithm with uncertainty input [3].

2.2.2 Balanced Comparison of Model Complexity and Likelihood

We now train $\psi_\kappa \in \Psi \; \forall\kappa$, i.e. train each system having a different number of symbols. Our goal is to select a system that can describe data well while having reasonable amount of complexity. However, in practice, a model with the lowest MDL score does not guarantee to be the best, as we need exhaustive dataset to compute ideal $P(M)$ and $P(D|M)$. Hence, there is often discrepancy between the value ranges of $-logP(M)$ and $-logP(D|M)$.

Generally, the model description length $-logP(M)$ changes in much higher amount than $-logP(D|M)$ if sampled data were obtained from the same domain, which makes $-logP(M)$ "dominate" MDL score. Hence, it is common practice to adjust both terms of MDL by multiplying weights to eliminate the biasing problem, but the result relies on the weights. However, although we do not know the value ranges of the two MDL terms, for sure if both $-logP(M)$ and $-logP(D|M)$ are less than that of another model, it is a better model. This is same as finding the Pareto-optimal solutions.

From this observation, we propose a balanced comparison method. First, while performing SCFG learning algorithm which searches for the best model of a system ψ by incrementally changing model parameters, save a pair of MDL components $p = [-logP(M), -logP(D|M)]$ at each step. We obtain these values from all systems $1...\kappa$ and call this set $S = \{p_1, p_2, ...p_n\}$. Compute S^*:

$$S^* = S - \Phi(p_i, p_j) \quad \forall i, j \tag{6}$$

where

$$\Phi(p_i, p_j) = \begin{cases} p_i & \text{if } p_i \succ p_j \\ \phi & \text{otherwise} \end{cases} \tag{7}$$

and $p_i \succ p_j$ is true only if both components of p_i are larger than p_j, respectively. We vote on S^* how many points belong to each model ψ_κ and choose N-best models. We have now obtained a candidate of models that can represent an activity effectively.

3 Experiments and Analysis

We set our objective to be imitation learning where a robot observes human demonstrator and follow a sequence of actions. Instead of simply imitating, we require that it should deal with observation error using the obtained knowledge so that it can correctly perform the intended action sequence. Furthermore, we consider that the task includes recursion which can be demonstrated in various lengths of action sequences, resulting in a more challenging setting.

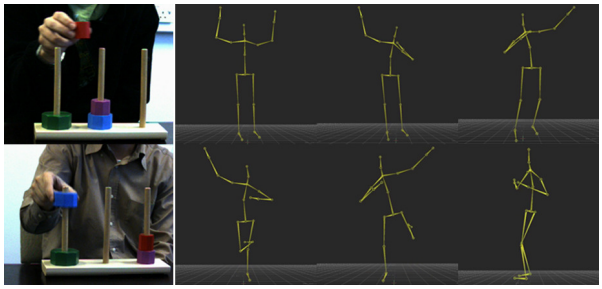

Figure 2. Example input data of the towers of Hanoi (left) and Dance (right) experiment.

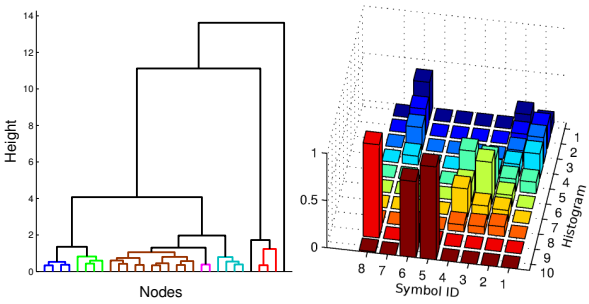

Figure 3. An example clustering tree created (left), showing only the top 30 nodes for better view, and eight symbols (right) obtained in the towers of Hanoi experiment.

We apply our method on two types of data: 1) video data of person solving the towers of Hanoi, and 2) motion capture data of person dancing. We use the dataset reported in [7] for 1) which includes 30 video clips of 5 participants solving the puzzle captured in 640x480, 30 fps. We implement a low-level tracker which tracks any moving blocks where a video segment is represented as a 10-dimensional vector which includes the tracker's x, y positions and frame differences dx and dy (velocities). We captured a new dataset for 2) to include dancing movements with recursion using an OptiTrack 8-camera system in 100 Hz. It consists of 25 demonstrations in total. Similar to [15], 6 most informative joints are selected for learning which makes our segments to be 6 dimensional vectors. Sample data can be seen on Fig. 2.

We first analyze the first dataset, the towers of Hanoi.

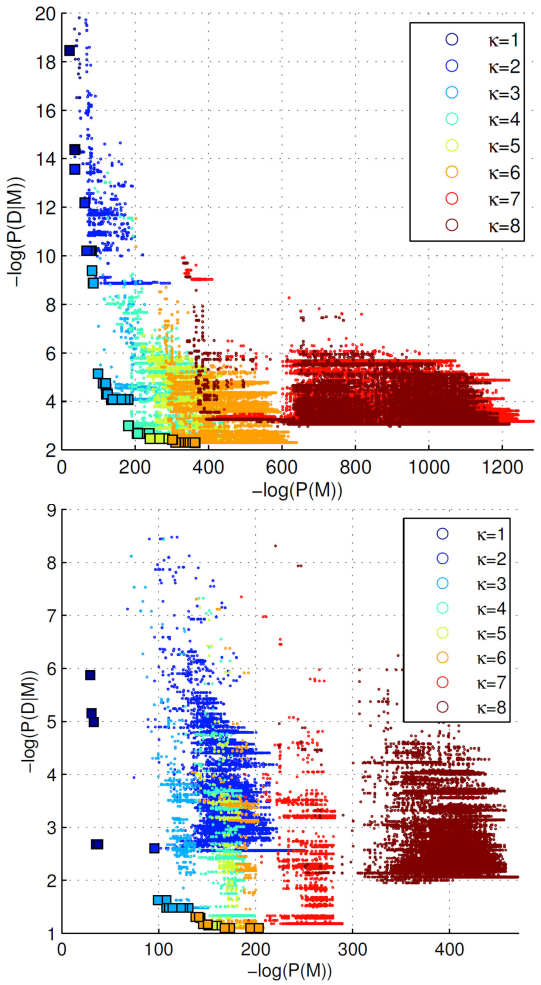

Figure 4. Spanning values of description lengths obtained from the towers of Hanoi (top) and Dance (bottom) data. Best cases (S^*) obtained using the method described in Sec. 2.2.2 are indicated by square markers. (Best viewed in color.)

The optimal solution to solve the puzzle requires 5 symbols, which respectively represent a disk to be lifted, placed, and moved between two out of three towers (3 symbols in total). Fig. 3 shows an example tree constructed and symbol representations with $\kappa = 8$.

Fig. 4 shows the spanning values obtained while inducing a grammar for each system ψ_κ. As can be seen, the likelihood does not improve as the number of symbols increases, because the learned model often fails to capture the regularity due to excessive number of symbols. The voting scores in Table 1 suggest that systems ψ_3 and ψ_5 are selected as the best.

This is reasonable since the towers of Hanoi puzzle can be also represented using 3 symbols, in which case they are interpreted as: "Disk lifted", "Disk dropped", "Disk transferred". However, this is not sufficient to actually solve the puzzle, as the symbol "Disk transferred" is ambiguous, i.e. it only describes *any* movement between two towers. Its representation is actually an averaged histogram of 3 different block transfer actions between two towers, which lacks specificity for execution. This is why systems having 5 symbols failed completely. Our method explicitly takes into account the problem of defining the right "scale" (scope) of a single action, which is generally problem-dependent.

Table 1. Results on the *towers of Hanoi* (**T**) and *Dance* (**C**) dataset. α and β denote mean $\pm$ standard deviation of $-logP(M)$ and $-logP(D|M)$, respectively. Votes (**V**) are computed by the method described in Section 2.2.2, whereas success rates (**S**) are computed by comparing the parsed symbols.

κ	α_T	β_T	V_T	S_T	α_C	β_C	V_C	S_C
1	45.3±7.5	16.3±2.0	2	0.00	34.5±4.1	4.0±1.5	5	0.00
2	142.0±42.4	10.1±1.8	6	0.00	177.4±20.4	3.3±0.9	1	0.00
3	202.5±30.8	4.2±0.6	**15**	0.00	124.6±10.9	2.9±1.1	**10**	0.00
4	319.3±43.1	3.0±0.9	8	0.00	173.7±12.0	2.5±1.0	0	0.00
5	356.6±38.2	2.8±0.7	**13**	**0.92**	172.4±8.1	2.5±1.0	4	0.00
6	463.0±58.2	2.5±0.6	9	0.50	191.1±10.7	2.2±1.1	**8**	**0.95**
7	925.3±133.7	3.3±0.4	0	0.92	259.1±13.5	2.0±0.8	0	0.95
8	947.4±114.6	3.1±0.3	0	0.67	413.2±20.8	2.3±0.5	0	1.00

To validate, we parse the input data using the obtained grammar of each system and execute to reproduce actions. During execution, each parsed symbol is mapped to the closest executable action, i.e. one of the five possible movements mentioned above. As the rule of the puzzle enforces that only a smaller disk shall be placed on top of a bigger disk, there is always only a single possibility of moving a disk between two towers. This is a fair assumption as this rule is always given in prior, not something to be learned. It is marked as success only if the parsed symbols lead to solve the puzzle. ψ_5 showed to be the best considering both success rate and the number of votes, which coincides with the ideal number of symbols.

The Dance dataset is composed of 6 motion primitives (a-f): Raise right or left arm (a, b), Raise both arms (c), Lift left or right leg while raising left or right arm, respectively (d, e), Spin 360° (f). Dance movements are represented as $(abc)^n(def)^n$, where $n = \{1, 2, 3\}$ in our dataset. (See Fig. 2) The result is shown in Fig.4. The execution is marked as success only if the parsed symbols exactly match the performed motion primitives.

Note that Fig. 4 is computed *without* any knowledge about the success condition, i.e. success rates are used only to verify the validity of the voting results.

The sample grammars learned from the Dance dataset are shown in Fig. 5. As stated above, it was originally demonstrated using 6 symbols. Fig. 5(a) shows

S→SEAB [0.399853]
| CFD [0.372613]
| SSEAB [0.199826]
| CEA [0.027708]

(a) 6 symbols (ideal)

S→CDD [0.416571]
| AEBS [0.389128]
| AEBSS [0.179596]
| ACBACDSS [0.014705]

(b) 5 symbols

S→SS [0.347360]
| BCD [0.294369]
| EGF [0.263985]
| SSSS [0.063192]
| EAC [0.020580]
| SEAFSS [0.010513]

(c) 7 symbols

Figure 5. Example grammars learned from data. (a) A grammar generated by a system ψ_6 having 6 symbols A-F. (b) has 1 less symbol, where one of the symbols represents two different actions. (c) has 1 more symbol, where the same action could be represented with two different symbols. Low-probability rules ($< 3\%$) exist due to input data noise.

the learned grammar with the ideal number of symbols, which are internally represented as A-F. Fig. 5(b) shows the case where the system lacks one symbol. As a result, the algorithm needs to reuse one of the symbols to represent 2 actions which are the most similar to each other relative to other actions. In contrast, Fig. 5(c) shows a grammar represented with 7 symbols, where two symbols could be used to execute the same action. Due to the noise inherent in captured data, there are some erroneous rules having less than 3% rule probabilities.

4 Conclusions

In this paper, we have presented an unsupervised method of selecting models with the "right" number of action symbols. We use hierarchical agglomerative clustering analysis and Pareto-inspired voting principles to tackle the balancing problem that commonly occurs in MDL score computations. It takes into account the question of choosing the right scope of a single action, which is generally problem-dependent.

Our method exploits the outcomes of SCFG learning technique as feedback to tune the number of symbols, where both grammar learning and symbol discovery are done in unsupervised way. The results confirm that our method is capable to discover and learn the optimal set of action symbols correctly.

The result of the towers of Hanoi shows an interesting aspect where the proposed method captured the 2 most reasonable models, ψ_5(ideal) and ψ_3, with notable distinction compared to others. Similarly, in the Dance dataset, ψ_6(ideal) and ψ_3 were chosen, which are also reasonable candidates. The results were obtained without any prior knowledge about the success criteria.

Acknowledgments

This work was partially supported by the EU FP7 projects ALIZ-E (FP7-ICT-248116) and EFAA (FP7-ICT-270490).

References

[1] J. Aggarwal and M. Ryoo. Human activity analysis: A review. *ACM Computing Surveys*, 43(3):16, 2011.
[2] Y. Demiris and B. Khadhouri. Hierarchical attentive multiple models for execution and recognition of actions. *RAS*, 54(5), 2006.
[3] Y. Ivanov and A. Bobick. Recognition of visual activities and interactions by stochastic parsing. *TPAMI*, 2000.
[4] K. Kitani, S. Yoichi, and A. Sugimoto. Recovering the basic structure of human activities from noisy video-based symbol strings. *IJPRAI*, 2008.
[5] D. Kulic, W. Takano, and Y. Nakamura. Combining automated on-line segmentation and incremental clustering for whole body motions. *ICRA*, 2008.
[6] P. Langley and S. Stromsten. Learning context-free grammars with a simplicity bias. In *ECML*, 2000.
[7] K. Lee, T.-K. Kim, and Y. Demiris. Learning reusable task representations using hierarchical activity grammars with uncertainties. In *IEEE International Conference on Robotics and Automation*, 2012.
[8] Y. Liang, S. Shih, A. Shih, H. Liao, and C. Lin. Learning atomic human actions using variable-length markov models. *T. System, Man & Cybernetics, Part B*, 2009.
[9] J. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *IJCV*, 79(3):299–318, 2008.
[10] I. Ota, R. Yamamoto, T. Nishimoto, and S. Sagayama. On-line handwritten kanji string recognition based on grammar description of character structures. *ICPR*, 2008.
[11] K. Pastra and Y. Aloimonos. The minimalist grammar of action. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 367(1585):103–117, 2012.
[12] Z. Solan, D. Horn, E. Ruppin, and S. Edelman. Unsupervised learning of natural languages. *PNAS*, 2005.
[13] A. Stolcke and S. Omohundro. Inducing probabilistic grammars by bayesian model merging. *Gramm. Infer. and App.*, pages 106–118, 1994.
[14] S. Wong, T. Kim, and R. Cipolla. Learning motion categories using both semantic and structural information. *CVPR*, 2007.
[15] F. Zhou, F. Torre, and J. Hodgins. Aligned cluster analysis for temporal segmentation of human motion. In *Automatic Face & Gesture Recognition*, 2008.