Automated Interpretations of Unsupervised Learning-Derived Clusters for Activity Recognition

Richard G. Freedman[1] and Shlomo Zilberstein[1]

I. INTRODUCTION

For robots to properly interact with humans, it is important that they are able to recognize users' plans and activities so that they may respond accordingly. Many activity recognition (AR) algorithms involve signal processing [1] or supervised learning [2], [3] to label raw sensor data with a human-defined action, but these methods restrict the generalizability of the learned activity models. In particular, sensor inputs that drastically differ from the training data cannot be labeled correctly, making it hard to build robots that respond appropriately to novel actions. This is evident in the codebook method used by Wang and Mori [2] to cluster the set of all sensor readings into k clusters via k-means clustering of the training data. The signal input at the center of each cluster is used to represent all sensor readings that fall within the cluster. This means that any novel signal input is aliased with one from the training data so that a misclassified action/activity is certain. Such a phenomenon was observed by Zhang and Parker [4] when they derived a codebook for compressed vector representations of spatial-temporal input signals, but they used an unsupervised topic model for AR.

Unsupervised learning methods such as the latent Dirichlet allocation topic model (LDA) [5] were recently proposed for AR tasks which enable machines to derive their own activity clusters [6]. The primary challenge with such methods is the inability for humans to clearly interpret these machine-defined actions, which can lead to difficulty in verification and determining response behaviors. The original work by Huỳnh et al. [6] provided interpretive evidence by aligning the learned topics with an annotated timeline of activities, but few other applications have had access to such annotations. Furthermore, it is evident that machine learning of practical activity clusters can be difficult when displaying sensor readings since numbers are not always easy to relate. Freedman et al. [7] represented activity clusters learned using LDA on red, green, blue, depth (RGB-D) sensor data as collections of stick figures in an attempt to resemble the topic modeling literature where collections of words are presented for each topic. As snapshots of an activity in progress, stick figures still cannot reveal the underlying trend(s) between each other like actual words can because *words have semantic definitions*.

We thus propose the use of feature vectors for signal data such as postures read by a RGB-D sensor in order to autonomously derive descriptions of activity clusters learned using unsupervised methods. This will remove ambiguity in the learned models because the machine can explain the trends in terms that humans comprehend. After providing an example of deriving such feature vectors for RGB-D sensor data and using them to present descriptions of a learned activity, we discuss how this may be used to develop more robust resposnsive behaviors during human-robot interaction.

II. LABELING RGB-D SENSOR INPUT

To derive commonalities between sensor readings in an activity cluster, we must have a list of human-defined properties for each reading. We then define a ***feature descriptor*** for input v as binary vector $\overrightarrow{x_v} \in \{0,1\}^{|F|}$ where F is the list of possible features and $x_v(i) = 1$ if and only if v has the i^{th} feature. For describing activity cluster t's features, we define a ***weighted feature descriptor*** as vector $\overrightarrow{x_t} \in [-1,1]^{|F|}$ where $x_t(i) \to 1$ as the i^{th} feature is more common in the cluster's readings and $x_t(i) \to -1$ as it is less common.

A. GENERATING FEATURE DESCRIPTORS

RGB-D sensor data produces a sequence of three-dimensional point clouds which present a colored surface of the region facing the sensor over time. Each point cloud may be used in AR to represent the environment where regions of changing points over time indicate objects of interest [4], and human bodies may be identified from these regions [8] to extract postures independent of the environment [7]. When a person looks at a single posture, she is able to explain it in terms of the appendages and joints' relative positions. For example, Fig. 1 is standing with the arms slightly bent, one which is raised, and one lifted leg that is bent. The conditions for discerning these features are not arbitrary because *specific angles of orientation for each joint dictate the orientation and position of the limbs*. As most software packages provide RGB-D sensor data in the form of $[-\pi,\pi]^{45}$ (roll, pitch, and yaw for 15 joints), it is possible to compute Euler angles and determine these features using a list of conditional statements. For example, an elbow joint may be considered bent if the angle between the upper and lower arm is in $[0, 3\pi/4]$ and straight if it is in $(3\pi/4, \pi]$.

B. GENERATING WEIGHTED FEATURE DESCRIPTORS

After learning the unsupervised AR model, we will have k clusters which partition the sensor inputs from the training data. In the case of unsupervised topic models such as LDA, our inputs are documents $d \in \{1,\ldots,D\}$ (a sequence of sensor readings) whose attributes are its words $\overrightarrow{w_d}$ (sensor readings) so that we specifically learn a topic (activity cluster) assignment $\overrightarrow{z_d}$ for each reading. Then each sequence

This research was supported in part by NSF grant number IIS-1405550.

[1] College of Information and Computer Sciences, University of Massachusetts, Amherst, MA 01003, USA {freedman,shlomo}@cs.umass.edu

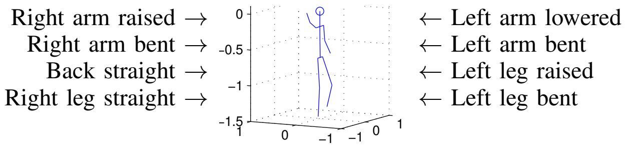

Fig. 1. Example of a feature descriptor for the pose above.

has a distribution of activities θ_d based on the ratio of assignments in $\overrightarrow{z_d}$ and each activity t has a distribution over readings ϕ_t based on the ratio of each pose v in all $\overrightarrow{w_d}$ assigned t:

$$\phi_t\left(v\right) = \frac{\left(\sum_{d=1}^{D}\sum_{n=1}^{\left|\overrightarrow{w_d}\right|}\mathbf{1}\left(w_d\left(n\right) = v \wedge z_d\left(n\right) = t\right)\right)}{\left(\sum_{d=1}^{D}\sum_{n=1}^{\left|\overrightarrow{w_d}\right|}\mathbf{1}\left(z_d\left(n\right) = t\right)\right)}$$

where $\mathbf{1}$ is the indicator function that equals 1 when the condition is true and 0 otherwise. Smoothing is usually applied based on some hyperparameter settings as well.

Because each θ_d is easily interpreted as a mixture of activities, we are most interested in finding interpretations for each ϕ_t because the relationships between sensor readings (poses) are not often as obvious. From the AR perspective, we want to identify which features best describe the majority of the sensor readings represented by each cluster's learned distribution. Using feature descriptors, we propose three approaches for computing a weighted feature descriptor:

1) Center of Mass: Let us consider each possible sensor input v as an independent particle in space with mass $\phi_t\left(v\right)$ and position $\overrightarrow{x_v}$. Then each particle is located at some corner of this $|F|$-dimensional space and has mass proportional to its probability density. The center of mass for a system of particles is the weighted average position between all the particles: $\overrightarrow{x_t} = \sum_{v=1}^{|\phi_t|}\phi_t\left(v\right)\cdot\overrightarrow{x_v}$ where the normalizing constant would be 1, the cumulative distribution over ϕ_t. Although simple to compute, this approach is naïve because it simply finds the weighted union of features. Thus a single sensor input with a large mass would contribute all its features to the cluster's weighted feature descriptor even if no other inputs with considerable mass share some of them.

2) Agglomerative Clustering: Agglomerative clustering hierarchically builds sets of objects that share like features, beginning with singleton sets that contain each sensor input. The likeness between sets C_1 and C_2 for cluster t is measured using $d\left(C_1, C_2\right) = \left|\sum_{v\in C_1}\phi_t\left(v\right) - \sum_{v\in C_2}\phi_t\left(v\right)\right| \cdot ||\overrightarrow{x_{C_1}} - \overrightarrow{x_{C_2}}||_2$ where $\overrightarrow{x_{C_i}}$ is the weighted feature descriptor for set C_i. d is not a metric because a distance of 0 does not guarantee that the two sets are equal. At each iteration, the set(s) with the smallest distance between each other are merged together as a new set; this process terminates when the distances are all greater than some threshold. While these distanced sets' weighted feature descriptors may be joined by weighted union like in the particle system above, the sensor inputs within each set are joined by the intersection of features: $\overrightarrow{x_{C_{1,2}}} = \sum_{v\in C_1\cup C_2}\phi_t\left(v\right)\cdot\bigodot_{v\in C_1\cup C_2}\overrightarrow{x_v}$ where $\odot$ is element-wise multiplication. Intersection may be too strong since it has the opposite problem of the union: a single sensor input with a large probability density may not have one feature that the remaining inputs of significant probability share. To address this, we introduce a ***soft intersection*** which accounts for the number of inputs sharing the presence/lack of a feature. We first convert each feature descriptor into $x'_v\left(i\right) = -1^{1+x_v\left(i\right)}$ and then compute $\overrightarrow{x_{C_{1,2}}} = \sum_{v\in C_1\cup C_2}\phi_t\left(v\right)\cdot\left(|C_1| + |C_2|\right)^{-1}\cdot\sum_{v\in C_1\cup C_2}\overrightarrow{x'_v}$.

3) Supervised Learning: The last approach acknowledges the fact that supervised learning methods such as decision trees learn interpretable functions. If we consider every sensor input in every recording sequence as a separate data point, then we have inputs $\overrightarrow{x_{w_d(n)}}$ with assigned outputs $z_d\left(n\right)$ from our AR model. We may use off-the-shelf supervised learning algorithms to learn a function mapping between each feature descriptor and its associated topic. The only limitation is that each algorithm has a specific type of function which it can learn. For example, decision trees can only learn perpendicular partitions of the space.

III. Future Work

We are currently implementing the generators described in Section II so that we may compare how well each approach performs. It is necessary that the derived interpretations not only make sense, but match a human's so that programmed response behaviors are appropriate. Furthermore, clearer breakdowns of actions perceived by the robot will enable more robust interaction since the perceived actions will have specific features such as which limbs are used. This will enable a robot to do motion planning with respect to these features rather than just the generic action; for example, a robot may plan to receive an item from the user's left hand rather than wait for the person to bring the item to it. Besides applying this to AR, we will also investigate whether it applies to a natural language analogue of topic modeling. This could further assist the field of human-robot interaction by improving semantic interpretation in dialogue systems when humans use synonyms or describe object features.

References

[1] R. Kelley, M. Nicolescu, A. Tavakkoli, C. King, and G. Bebis, "Understanding human intentions via hidden Markov models in autonomous mobile robots," in *Proc. of the 3rd ACM/IEEE Int'l Conf. on Human-Robot Interaction*, 2008, pp. 367–374.

[2] Y. Wang and G. Mori, "Human action recognition by semi-latent topic models," *IEEE Transactions on Pattern Analysis and Machine Intelligence Special Issue on Probabilistic Graphical Models in Computer Vision*, vol. 31, no. 10, pp. 1762–1774, 2009.

[3] H. S. Koppula and A. Saxena, "Learning spatio-temporal structure from RGB-D videos for human activity detection and anticipation," in *Proc. of the Int'l Conf. on Machine Learning*, 2013, pp. 792–800.

[4] H. Zhang and L. E. Parker, "4-dimensional local spatio-temporal features for human activity recognition," in *Proc. of the IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, Sept. 2011, pp. 2044–2049.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[6] T. Huỳnh, M. Fritz, and B. Schiele, "Discovery of activity patterns using topic models," in *Proc. of the 10th Int'l Conf. on Ubiquitous Computing*, 2008, pp. 10–19.

[7] R. G. Freedman, H.-T. Jung, and S. Zilberstein, "Plan and activity recognition from a topic modeling perspective," in *Proc. of the 24th Int'l Conf. on Automated Planning and Scheduling*, 2014, pp. 360–364.

[8] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from a single depth image," in *Proc. of the 24th IEEE Conf. on Computer Vision and Pattern Recognition*, 2011, pp. 1297–1304.