Action Recognition from Skeleton Data via Analogical Generalization over Qualitative Representations

Kezhen Chen and Kenneth D. Forbus
Northwestern University
KezhenChen2021@u.northwestern.edu

Abstract

Human action recognition remains a difficult problem for AI. Traditional machine learning techniques can have high recognition accuracy, but they are typically black boxes whose internal models are not inspectable and whose results are not explainable. This paper describes a new pipeline for recognizing human actions from skeleton data via analogical generalization. Specifically, starting with Kinect data, we segment each human action by temporal regions where the direction of motion is constant, creating a *sketch graph* that provides a form of qualitative representation of the behavior that is easy to visualize. Models are learned from sketch graphs via analogical generalization, which are then used for classification via analogical retrieval. The retrieval process also produces links between the new example and components of the model that provide explanations. To improve recognition accuracy, we implement dynamic feature selection to pick reasonable relational features. We show the explanation advantage of our approach by example, and results on three public datasets illustrate its utility.

Introduction

Human action recognition is an important but difficult problem. Traditional machine learning relies on extracting large numbers of features and using techniques such as deep learning (Baccouche et al. 2011). However, these techniques have some disadvantages. A key problem is that they are black boxes: They can produce results, but do not provide explanations for their answers. This makes their results difficult to trust and to debug (Lowd and Meek. 2005). When people perform recognition, even for visual tasks, they often can describe the reasons for their classification. There is evidence that relational representations are important in human cognition (Marr 1982; Palmer 1999). By working with human-inspired relational representations, we provide evidence that analogical models can produce high accuracy while providing explanations.

This paper draws on research in qualitative spatial reasoning and cognitive simulation of visual problem-solving and analogy to provide a new approach with high accuracy and a novel explanation ability to recognize human actions from Kinect skeleton data. Instead of computing frame-based features (Wang et al. 2016; Li, Chen, and Sun. 2016), the video stream is divided into *sketch graphs*, consisting of multiple sequences of snapshots. Each snapshot is like a panel in a comic strip: It consists of a motion segment described by a single qualitative state, which might correspond to many frames. Each body point has its own sequence of such states. The trajectories within these states and relationships across these states are described qualitatively, using automatically constructed visual representations. The sketch graphs for each instance of a behavior type are combined via analogical generalization, to automatically construct probabilistic relational schemas (plus outliers) characterizing that behavior type. To categorize a new behavior, a set of sketch graph representations is computed for it, and analogical retrieval is used across the entire set of trained behavior models to retrieve the closest schema (or outlier). We begin by summarizing the work we build on, including CogSketch and analogical processing. We then describe the learning pipeline and how classification works. We show how *explanation sketches* enable understanding recognition decisions made via analogy. Results on three public Kinect action datasets are described, and we close with related and future work.

Background

Our approach combines ideas from sketch understanding and analogical processing. We discuss each in turn.

CogSketch

CogSketch (Forbus et al. 2011) is a sketch understanding system that provides a model of high-level visual processing. It provides multiple, hierarchical levels of visual representation, including decomposing digital ink into edges, combining edges into entities, and gestalt grouping methods. The qualitative visual representations that it automatically computes from digital ink have enabled it to model a variety of visual problem-solving tasks (e.g. Lovett and Forbus, 2011). These relations include qualitative topology (Gohn et al. 1997), positional relations (e.g. above, leftOf), and directional information (e.g. quadrants). We extended the OpenCyc ontology with these relations. Every sketch has one or more subsketches, each of which contains glyphs. Glyphs are the constitutes of sketches. Subsketches can participate in relationships.

The representations produced by CogSketch have been used to model human performance on several visual tasks, including Ravens' Progressive Matrices, one of the most common tests used to measure human fluid intelligence. The CogSketch model uses SME (described below) at multiple levels of visual representations, including re-representing visual descriptions automatically as needed. Its performance places it in the 75th percentile for American adults, better than most adult Americans (Lovett and Forbus, 2017). Ravens and the other visual problems that CogSketch and SME have been used with are static, this paper marks the first time they have been used with dynamic visual data.

In this paper, subsketches are used to implement sketch graphs. CogSketch's visual processing is used to construct additional relations within and between subsketches. This includes the qualitative representations mentioned above. In each subsketch, relations within an action segment are extracted. The *metalayer* in CogSketch enables multiple subsketches and relationships between them to be displayed, to support visualization.

Analogical Processing

We build on models inspired by Gentner's structure-mapping theory of analogy and similarity (Gentner, 1983). Its notion of comparison is based on structured descriptions, including both attributes and relations. There is considerable psychological evidence supporting structure-mapping, making it attractive for use in AI systems so that, with the right representations, what looks similar to us will look similar to our software and vice-versa. We use Structure-Mapping Engine (SME; Forbus et al. 2016) for analogical matching, MAC/FAC (Forbus, Gentner, and Law. 1995) for analogical retrieval, and SAGE (McLure, Friedman and Forbus. 2015) for analogical generalization. Since these operations are at the heart of our learning approach, we summarize each in turn.

SME takes as input two structured, relational representations and produces one or more *mappings* that describe how they align. These mappings include correspondences (i.e. what goes with what), a similarity score, and candidate inferences that suggest how statements from one description can be projected to the other. SME has been used in a variety of AI systems and cognitive models.

Analogical retrieval is performed by MAC/FAC, which stands for "Many are Called/Few are Chosen", because it uses two stages of map/reduce for scalability. The inputs consist of a probe case and a case library. The MAC stage computes, in parallel, dot products over vectors that are automatically constructed from structured descriptions, such that each predicate, attribute, and logical function are dimensions in the vector and whose magnitude in each dimension reflects their relative prevalence in the original structured description. The best mapping, and up to two others (if they are sufficiently close) are passed to the FAC stage. FAC compares the best structured descriptions from the MAC stage to the input probe using SME. Again, the best match, with up to two others if sufficiently close, are returned. This provides scalability (because the MAC stage is inexpensive) as well as structural sensitivity (because the content vector dot product is a coarse estimate of SME similarity, followed by using SME itself).

Analogical generalization is performed by the Sequential Analogical Generalization Engine (SAGE). Every concept to be learned by analogy is represented by a *generalization pool*, which maintains both generalizations and outlying examples. Examples are added incrementally. The closest matching item (example or generalization) is retrieved via MAC/FAC, using the contents of the generalization pool as a case library. If there is no item, or the similarity to what is retrieved is less than an *assimilation threshold*, the new example is added as an outlier. Otherwise, if the item retrieved is an example, the two are combined into a new generalization. This process involves merging them, replacing non-identical entities by skolems, and assigning a probability to each statement depending on whether it was in just one description or both. If the item retrieved was a generalization, that generalization is updated with skolems and probabilities based on its alignment with the new example. Generalizations in SAGE are thus probabilistic, but still concrete – skolem entities may become more abstract due to fewer high-probability statements about them, but logical variables are not introduced. Instead, candidate inferences are used for schema application.

SAGE also supports classification, by treating the union of generalization pools as a large case library. The case library which contained the closest item is taken as the classification of that example, with the correspondences of the match constituting an explanation of why it is a good match. Since a generalization pool can have multiple generalizations, SAGE naturally handles disjunctive concepts.

Our Approach

Our approach focuses on human skeleton action recognition via analogical generalization over qualitative representations. It is implemented as a pipeline with four stages: Action Segmentation, Relational Enrichment, Action Generalization and Classification. A dynamic feature selection process picks reasonable additional features for different actions before the final training. All sketches and relations are computed from our system automatically. Figure 1 shows the pipeline of our system. We describe each stage in turn, and describe explanation sketches.

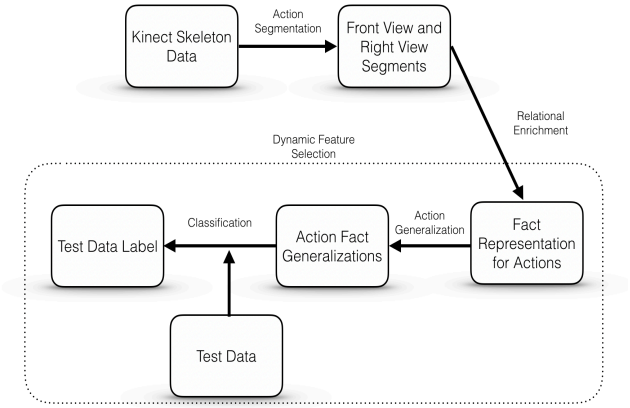

Figure 1: pipeline of our algorithm

Action Segmentation

The skeleton data produced by a Kinect (or other 3D sensors) contains many points per frame, representing each body part such as the head or right-hand. We use 20 body points tracked by Kinect V1 to represent 20 body parts and connect these body points to provide a concise body skeleton graph, as shown in Figure 2.

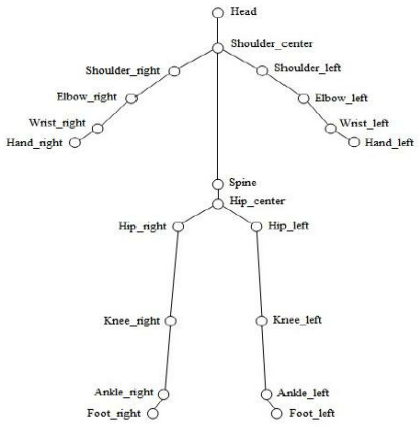

Figure 2: Kinect body skeleton graph

Each instance of an action consists of a continuous movement stream, sampled via many frames, each frame containing coordinates for these points. The first step of our pipeline abstracts away from frames into qualitatively distinct intervals describing the motion of particular body parts. A *track* is a sequence of point coordinates from each frame for a specific body point. As CogSketch needs 2D sketches, we map each 3D coordinate into front-view and right-view. To segment movements of a track (a body point) in a view, we compute the *azimuth* (the angle that is clockwise relative to the north) changes of the track frame by frame to find the direction change. Intervals of time over which the motion has similar azimuth are grouped into one segment. In the experiments reported here, we use only the right-hand, left-hand, right-hip and left-hip in front-view and right-view, because the motions in the datasets used can be described as the movements of these four body parts.

For each track in a view, we first compute the spatial relation *Moving or Stationary* (MOS), with 0.02 quantization factor via *QSRlib* (Gatsoulis et al. 2016), a library of qualitative spatial relations and calculi, for the four main body points. MOS relations can show whether a point in a frame is moving (label 'm') or stationary (label '0'). An MOS relation sequence could be as follows:

[0,0,0,0,m,m,m,m,m,m,m,0,0,]

After motion detection, eight MOS sequences corresponding to four points in two views are extracted. All frames with label 'm' are segmented by computing the cartographical azimuth changes between each pair of two consecutive moving points. When the azimuth change between two consecutive point pairs is larger than 88 degrees, the movement of the track is segmented into two parts. After action segmentation, we get eight sequences of segments, four points in the front-view and four points in the right-view.

We use two techniques to reduce segmentation noise. First, after action segmentation, all segments in a track are merged again when the average azimuth between start-point and end-point is smaller than fifty degrees. Second, segments are also merged when the distance between start-point and end-point of the segment is smaller than half of average distance between start-point and end-point of all segments.

Relational Enrichment

The relational enrichment stage involves automatically adding additional relationships via CogSketch, to provide more information about the motions within and between segments. Each example of a behavior is imported into CogSketch as a set of sketches, one per track, with each panel (segment) within a track being represented by a separate subsketch. Within each panel, the skeleton is represented by a set of glyphs, including an arrow from start-point to end-point of the track panel to represent the direction of motion. Figure 3 shows two sketches from eight sketches of raising the right-hand and putting it down and the relations within same sketch and between different sketches. As only the right-hand has movements in this action, we only show the two sketches of right-hand movements.

To summarize, each action is represented by eight sketches in CogSketch: right-hand in front-view, right-hand in right-view, left-hand in front-view, left-hand in right-view, right-hip in front-view, right-hip in right-view, left-hip in front-view and left-hip in right-view. In each sketch and subsketch, CogSketch is used to compute relationships between body parts, e. g. the relative position of the right-hand to the head.

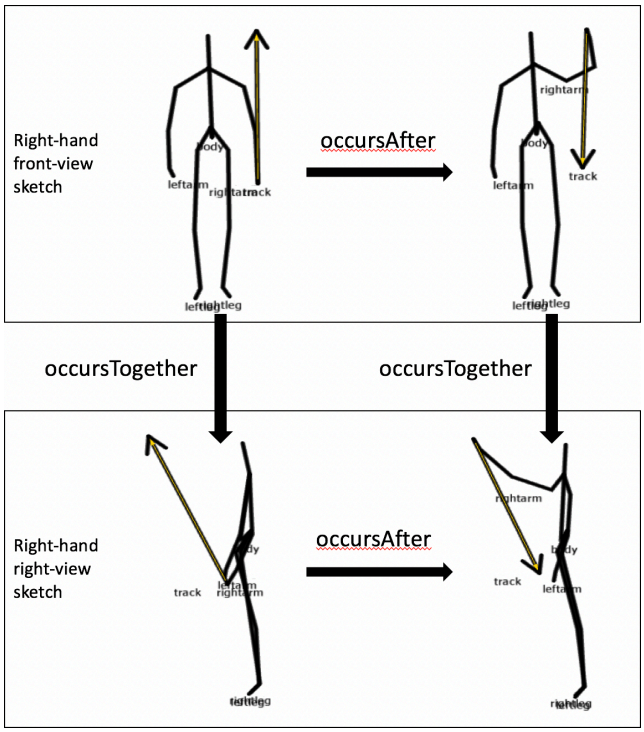

Figure 3: Two sketches of raising hand action

We use the following logical function to denote panels in a sketch graph:

(KinectMotionFn <body-part> <view> <move-type> <token>)

<body-part> is from the four main body points: right-hand, left-hand, right-hip, left-hip. <view> is front or right view. <move-type> describes the type of movement: single-hand, two-hand or full-body. <token> is a unique identifier denoting the segment.

In each segment subsketch, additional details can be provided via Cyc's holdsIn relation. For example, spatial relations are helpful to determine the locations of body points, so these relations are added. In each segment motion, we use entities from CogSketch's qualitative representation of orientation to describe the direction of motion, i.e. the quadrant representations Quad1, Quad2, Quad3, Quad4, the pure directions Up, Down, Left, and Right, plus the constant NoMotion indicating lack of motion. The motion direction information is connected with the motion segment via holdsIn. For example:

(holdsIn
(KinectMotionFn RightHand Front SingleHand D1RHFS2)
(trackMotionTowards Quad1))

Sequence information between segment panels is represented using the occursAfter and occursTogether relations. occursAfter indicates that two segments occur successively and is used to connect segments from same track in same view. occurTogether means that two segments have eighty percent time overlap and connects the segments from different track in same view, e. g.

(occursAfter
(KinectMotionFn RightHand Front SingleHand D1RHFS2)
(KinectMotionFn RightHand Front SingleHand D1RHFS4))

This representation enables facts from different segments to be included in one unified case representation and is extracted totally automatically. Table 1 provides the full set of relationships that we compute for every track.

Relations	Descriptions
(trackMotionTowards <Quad1/Quad2/Quad3/Quad4 >)	The track moving direction from start-point to end-point.
(quadDirBetween <right-hand/left-hand> <right-elbow/left-elbow> <Quad1/Quad2/Quad3/Quad4>)	The elbow direction with respect to corresponding hand.
(bodyStatus <Bend/Straight>)	Whether the body bends larger than 45 degrees.
(handPosition <RaiseHand/PutDown > <right-hand/left-hand>)	Whether the hand bends raising larger than 90 degrees.
(armStatus <Bend/Straight > <right-arm/left-arm>)	Whether the arm bends larger than 90 degrees.
(motionRange <Large/Small>)	Whether the movement range is larger than half of body length.
(twoArmRela <Cross/NoCross>)	Whether the two arms are cross with each other.
(legStatus <Bend/Straight> <right-leg/left-leg>)	Whether the leg bends larger than 45 degrees.
(moveRespectArm <Inside/OutSide> <right-hand/left-hand>)	Whether the hand is moving towards inside of the arm or outside of the arm.
(distRespectBody <Large/Small>)	Whether the distance of x coordinates between hand and spine is larger than 25.

Table 1: basic features in each case

Action Generalization

All facts for each segment are combined as a case representing the entire action. Each such action instance is added to the generalization pool being used to learn that concept. For all experiments reported here, we used an assimilation threshold of 0.7. SAGE also uses a probability cutoff, i.e. if a fact's probability drops below this value, it is removed from the generalization. We used a probability cutoff of 0.2 in all experiments. Each action type is represented as a distinct generalization pool and all action type generalization pools are combined into a case library.

Classification

By treating the union of generalization pools for action types as one large case library, our system can classify new examples based on which library the closest retrieved item came from. Given a new case, MAC/FAC is used to retrieve the

closest generalization or outlier from the union of the generalization pools. The action label of the generalization pool it came from is assigned to the test instance.

Dynamic feature selection

As shown in Table 1, ten basic relations are extracted to represent each segment. However, representing the direction of motion more precisely relative to different reference points can be very useful. The relation qualDirBetween represents the direction of motion with respect to a reference point. Its first argument is the start or end point of the motion. Its second argument is the reference point. Its third argument is the direction. For example,

(holdsIn
(KinectMotionFn RightHand Front SingleHand D1RHFS1)
(qualDirBetween RightHand-StartPoint Head Quad4))

indicates that that the start-point of the motion D1RHFS1 is in the Quad4 direction, with respect to the head, in the first segment of the right-hand track within the front-view.

In these experiments we use head, shoulder-center, spine and hip-center as the possible reference points. Directions are described either in terms of quadrants or broad directions, i.e. Left/Right or Up/Down, each of which are the unions of two quadrants. For conciseness, we will abbreviate subsets of these representation via the template <reference point>-<direction type>, i.e. the statement above would be an example of Head-Quad.

Dynamic feature selection is used to select which families of direction representations are used for a dataset. Given the distinctions above, there are 12 families of qualDirBetween relations that can be computed. The algorithm starts with the basic set of features plus a single family of optional features, doing training and testing with each independently. The highest accuracy optional feature is retained. On subsequent rounds, the search is constrained by limiting it to the two unused features that perform best where the choices so far perform the worst. The search stops either when a cutoff is reached (here, the cutoff is four optional features, which provides a reasonable tradeoff between accuracy and efficiency) or when all the additions lead to lower accuracy.

We evaluated dynamic feature selection on the Florence 3D Action dataset (Seidenari et al. 2013), which contains nine activities: wave, drink from a bottle, answer phone, clap, tie lace, sit down, stand up, read watch, bow. Ten subjects were asked to perform the nine actions two or three times. Two groups of additional features are tested: one is picked manually and the other one is picked via the algorithm above. Cross-subject validation was used. The results, shown in Table 2, show that dynamic feature selection improves accuracy by ten percent.

Methods	Features	Accuracy results (%)
Manual-feature-selection	Head-Quad, Spine-Quad, Hip-Center-Quad	63.6
Dynamic-feature-selection	Head-Quad Spine-Up-Down Hip-Center-Up-Down	74.2

Table 2: Recognition results with dynamic-feature-selection

Explanation Sketches

Sketch graphs carve motions up into a discrete set of snapshots, much like comic strips, an easy to apprehend visualization for people. The generalizations are also sketch graphs, enabling them to be inspected more easily than, for example, large vectors. The analogical mapping that justifies a recognition decision can be displayed by drawing the correspondences between panels in the two sketch graphs and their constituents.

This is the basis for the explanation sketch. It depicts how a retrieved sketch graph for a generalization (or an outlier) aligns with a sketch graph for a new behavior. The skeleton glyphs from corresponding segments are visualized in two boxes side-by side. Correspondences within the segments are indicated by dashed lines. Thus, the explanation sketch provides a visualization for how a model explains a behavior, based on their overlap. Figure 4 shows two examples. Figure 4 (a) shows a perfect match with five dashed lines for all five different parts. Figure 4 (b) shows a different movement corresponding, which only has same facts for left-arm, left-leg and right-leg.

The explanation sketch could be used to answer why-not question. Given a behavior and asked why a specific classification was not used, the most similar generalization or outlier from that generalization pool can be retrieved and compared. The non-overlapping aspects provide the gist for why that category was not chosen.

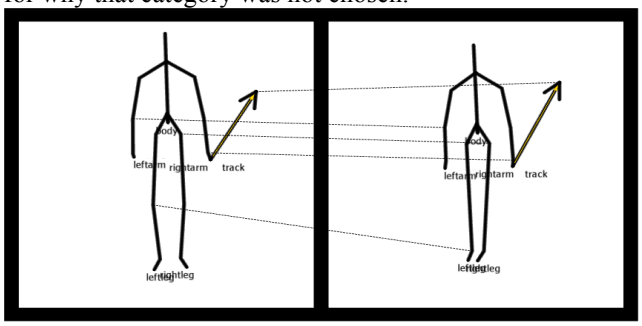

Figure 4 (a): perfect matching

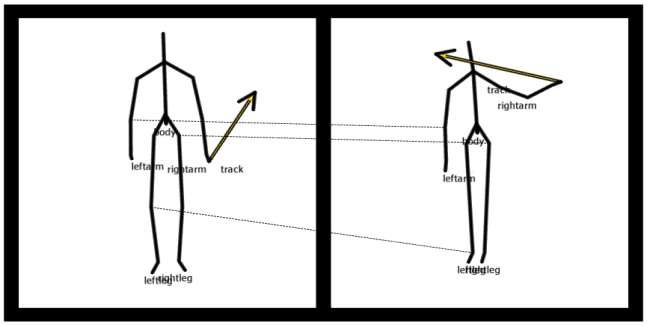

Figure 4 (b): Partial matching

Experimental Results and Discussion

While we view the ability to produce understandable explanations as an important part of our approach, we note that

other approaches do not explore explanation, so we confine ourselves here to comparing with others using their metrics. Three datasets are tested and we describe each in turn.

UTD-MHAD Dataset

The University of Texas at Dallas Multimodal Human Action Dataset (UTD-MHAD) was collected as part of research on human action recognition by Chen et al. (Chen et al. 2015). This dataset contains eight different subjects (4 females and 4 males) performing twenty-seven actions in a controlled environment. Data was collected using Kinect V1 sensor with 4 repetitions per action.

Qualitative spatial relations are computed for all target body points and dynamic feature selection is used. From dynamic feature selection, Head-Quad, Head-Up-Down, Hip-Center-Up-Down and Spine-Quad were picked as four additional features. We used the same cross-subject testing method from (Chen, et al ,2015) in our experiments. Our method achieves 65.82% accuracy. One reason why our method does not have relatively good performance on this dataset is that there are many similar actions in this dataset such as "Arm-curl" and "clap". Some pairs of similar actions have the same qualitative representations, so SME has trouble recognizing them. Consequently, we also test our method on a subset of actions containing nineteen actions (we remove 6 similar actions and 2 actions with large noise). Table 3 shows the accuracy for each action tested on the two different action sets. The average result of existing methods is in Table 4.

Accuracy (%) 27 actions			Accuracy(%) 19 actions	
Swipe left	68.75		Swipe left	75
Swipe right	62.5		Swipe right	68.75
Wave	81.25		Wave	81.25
Clap	56.25		Arm cross	68.75
Throw	43.75		Basketball shoot	81.28
Arm Cross	62.5		Draw circle Clockwise	87.5
Basketball shoot	81.25		Draw triangle	87.5
Draw X	50		Bowling	62.5
Draw circle Clockwise	75		Boxing	87.5
Draw circle Counter clockwise	25		Baseball swing	87.5
Draw triangle	68.5		Tennis serve	62.5
Bowling	62.5		Push	62.5
Boxing	87.5		Catch	81.25
Baseball swing	81.25		Pickup and throw	81.25
Tennis swing	37.5		Jog	100
Arm curl	50		Sit to stand	75
Tennis serve	62.5		Stand to sit	81.25
Push	56.25		Lunge	93.75
Knock	62.5		Squat	100
Catch	75		Overall	80.3
Pickup and throw	81.25			
Jog	62.5			
Walk	43.75			
Sit to stand	75			
Stand to sit	81.25			
Lunge	93.75			
Squat	100			
Overall	65.82			

Table 3: Accuracy (%) for each action from the two action sets

Method	Accuracy (%)
Inertial (Chen et al. 2015)	67.2
Kinect & Inertial (Chen et al. 2015)	79.1
CNN (Wang et al. 2016)	85.81
Our Method	65.82
Our Method (19 actions)	80.3

Table 4: Recognition Rates (%) comparison on the UTD-MHAD

As shown in the two tables above, our method has 80.3% accuracy on the set with 19 actions, which is only lower than the CNN approach. Again, one reason is that the qualitative relation encoding can cause information loss if the available relationships do not provide fine enough distinctions. For examples, the "swipe-right" action can be segmented into three parts: raise hand, swipe, and put down hand. But these segments could form a triangle in the air, which is similar to the "draw-triangle" action. With spatial relations we defined here, some instances of "draw triangle" with larger movement range may be represented by same relational facts of "swipe-right". This resolution/accuracy tradeoff is worth exploring in the future work.

Florence 3D Actions Dataset

We ran an experiment on this dataset but followed the leave one out cross-validation protocol (LOOCV) (Seidenari et al. 2013) to compare with their methods. With this protocol, there are more training data than the experiment in previous section. Dynamic feature selection picked Head-Quad, Spine-Up-Down, Hip-Center-Up-Down as additional features. The average accuracy compared with other methods is shown in Table 5.

Method	Accuracy (%)
Devanne et al. 2014	87.04
Vemulapalli, Arrate, and Chelappa. 2014	90.88
Our Method	86.9

Table 5: Recognition Rates (%) on the Florence dataset

As Table 5 shows, our method has comparable results with (Devanne et al, 2014) and a little lower than (Vemulapalli, Arrate, and Chelappa. 2014) results. In this dataset, our algorithm has relatively low accuracy on "drink from a bottle" and "answer phone" among all nine actions because some instances of them cannot be distinguished from the "wave" action. All three can be segmented into a motion that subject raises the right-hand to the position near the head, and our qualitative representations did not have sufficient resolution to distinguish them. However, we note that when people are asked to review the skeleton data for these two actions, they also find it hard to describe the differences. Consequently, we do not necessarily view our system's performance on these actions as a negative.

UTKinect-Action3D Dataset

To further evaluate our method, we ran an experiment on the UTKinect-Action3D dataset (Xia, Chen, and Aggarwal. 2012). This Kinect dataset has 10 actions performed by 10 different subjects. For each action, each person performs it twice so there are 200 behaviors in this dataset. The ten actions are: walk, sit down, stand up, pick up, carry, throw, push, pull, wave, and clap hands.

With dynamic feature selection, Head-Up-Down, Spine-Quad and Hip-Center-Up-Down are picked as the additional features for this dataset. We follow the leave one out cross validation protocol (LOOCV) for comparability. Table 6 shows the recognition rates corresponding to the different actions and compares our accuracy with two other methods.

Action	Xia, Chen, and Aggarwal. 2012	Theodorakopoulos et al. 2014	Ours
Walk	96.5	90	100
Sit down	91.5	100	90
Stand up	93.5	95	85
Pick up	97.5	85	100
Carry	97.5	100	85
Throw	59.0	75	60
Push	81.5	90	70
Pull	92.5	95	95
Wave	100	100	100
Clap hands	100	80	100
Overall	90.92	90.95	88.50

Table 6: Recognition Rates (%) on the UTKinect-Action dataset

As shown in the Table 6, the three algorithms present comparable performance for different actions. Our average accuracy is 88.5%. Our system has relatively lower accuracy on some actions such as throw and push. In this dataset, some subjects did not face the Kinect directly when they performed the actions. As our method needs to extract front-view and right-view sketches from the data, this noise could have influenced on our algorithm.

Related Work

Human action recognition from Kinect data is a popular topic and various methods have been used on this problem. In (Wang et al. 2016), the spatial-temporal information from 3D skeleton data was projected into three 2D images (Joint Trajectory Maps), and Convolutional Neural Networks were used for action recognition. In (Chen et al. 2015), three different depth motion maps of front, side and top views are extracted as features form depth video sequences. Additionally, each skeleton data sequence is partitioned into N temporal windows and some statistical features are extracted in each window. A collaborative representation classifier is used to classify the actions with the features above. In (Devanne et al. 2014), the skeleton data is modeled as a multi-dimensional vector and the trajectories described by this vector are interpreted in a Riemannian manifold. By using an elastic metric to compare the similarity between trajectories, actions can be classified. In (Vemulapalli, Arrate, and Chelappa. 2014), the skeleton data movements are represented as Lie group and mapped to its Lie algebra. Then, the authors test this representation with Fourier temporal pyramid representation and linear SVM. In (Xia et al. 2012), histograms of 3D joint locations with LDA are extracted from skeleton data and HMM models are trained to classify actions. In (Theodorakopoulos et al. 2014), the system uses sparse representations in dissimilarity space to encode action movements and performs classification on these representations.

All approaches mentioned above only use statistical models on quantitative data, which makes it hard for them to explain their results. To the best of our knowledge, this is the first work to do skeleton action recognition via analogical generalization on qualitative relations instead of pattern recognition or machine learning. We also note that none of the algorithms above address explanation, whereas our approach does. As far as we know, this is the first paper to provide visual explanations for skeleton action recognition problems. Admittedly, qualitative representations do lose details of action movements, so some methods have slightly better performance on accuracy than ours. However, explanation ability is also essential in recognition tasks, to provide people a better understanding of the results.

Conclusions and Future Work

This paper presents a new approach with high accuracy and novel explanation ability, based on qualitative representations and analogical generalization, for learning how to classify human actions from skeleton data. Our pipeline uses azimuth changes to segment tracks, a cognitive model of human high-level vision to enrich descriptions of motion and configuration, and analogical generalization to provide learning via inspectable, relational models. Explanation sketches are used to visualize the correspondences and mappings between different segments. Experiments on three public datasets provide evidence for the utility of this approach.

There are several avenues to explore next. The first is to test it with additional datasets, both to explore noise and dynamic encoding issues. The second is to explore the effectiveness of explanation sketches in helping system trainers improve performance and to implement the extension to SAGE, which constructs near-misses (McLure, Friedman and Forbus. 2015), to improve our explanation sketches. Furthermore, we plan to explore using this same approach to analyze video more broadly, including RGB and depth data.

Acknowledgements

This research was supported by the Machine Learning, Reasoning, and Intelligence Program of the Office of Naval Research.

Reference

Baccouche, M., Mamalet, F., Wolf, C., Garcia C., and Baskurt, A. 2011. Sequential Deep Learning for Human Action Recognition. Human Behavior Understanding, 29-39.

Chen, C., Jafari, R., and Kehtarnavaz, N. 2015. UTD-MHAD: A Multimodal Dataset for Human Action Recognition Utilizing a Depth Camera and a Wearable Inertial Sensor. Image Processing (ICIP), 2015 IEEE International Conference on (pp. 168-172). IEEE.

Cohn, A. G., Bennett, B., Gooday, J., and Gotts, N. M. 1997. Qualitative Spatial Representation and Reasoning with the Region Connection Calculus. GeoInformatica, 1(3), pp.275-316.

Devanne, M., Wannous, H., Berretti, S., Pala, P., Daoudi, M., and Del Bimbo, A. 2014. 3D human action recognition by shape analysis of motion trajectories on Riemannian manifold. IEEE transactions on cybernetics, 45(7), 1340-1352.

Duckworth, P., Gatsoulis, Y., Jovan, F., Hawes, N., Hogg, D.C., and Cohn, A.G., 2016. Unsuperivised Learning of Qualitative Motion Behaviours by a Mobile Robot. In Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems (pp. 1043-1051). International Foundation for Autonomous Agents and Multiagent Systems.

Forbus, K., Gentner, D., and Law, K. 1995. MAC/FAC: A Model of Similarity-based Retreval. Cognitive Science, 19(2), pp.141-205.

Forbus, K., Ferguson, R., Lovett, A., and Gentner, D. 2016. Extending SME to Handle Large-Scalle Cognitive Modeling. Cognitive Science. Cognitive Science, 41(5), pp. 1152-1201.

Forbus, K., Liang, C., and Rabkina, I. 2017. Representation and Computation in Cognitive Models. Topics in Cognitive Science, 9(3), pp.694-718.

Forbus, K., Usher, J., Lovett, A., Lockwood, K., and Wetzel, J. 2011. CogSketch: Sketch Understanding for Cognitive Science Research and for Education. Topics in Conitive Science, 3(4), pp. 648-666.

Gatsoulis, Y., Alomari, M., Burbridge, C., Doudrup, C., Duckworth, P., Lightbody, P., Hanheide, M., Hawes, N., Hogg, D.C. and Gohn, A.G. 2016. QSRlib: a software library for online acquisition of Qualitative Spatial Relations from Video. QR.

Gentner, D. 1983. Structure-mapping: A theoretical framework for analogy. Cognitive Science, 7(2), pp.155-170.

Kunze, L., Burbridge, C., Alberti, M., Tippur, A., Folkesson, J., Jensfelt, P., and Hawes, N. 2014. Combining Top-down Spatial Reasoning and Botton-up Object Class Recognition for Scene Understanding. In Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on (pp. 2910-2915). IEEE.

Li, J., Chen, J and Sun, L., 2016. Joint Motion Similarity (JMS)-based Human Action Recognition Using Kinect. In Digital Image Computing: Techniques and Applications (DICTA), 2016 International Conference on (pp. 1-8). IEEE.

Lovett, A., and Forbus, K. 2011. Cultural commonalities and differences in spatial problem-solving: A computational analysis. Cognition, 121(2), pp.281-287.

Lovett, A., and Forbus, K. 2017. Modeling visual problem solving as analogical reasoning. Psychological Review, 124 (1), p.60

Lowd, D., and Meek, C. 2005. Adversarial learning. In Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining (pp. 641-647). ACM.

Marr, D. 1982. Vision: A computation approach.

McLure, M., Friedman, S., and Forbus, K.D. 2015. Extending Analogical Generalization with Near-Misses. In AAAI. (pp. 565-571)

Palmer, S.E. 1999. Vision Science: Photons to Phenomenology. MIT Press.

Seidenari, L., Varano, V., Berretti, S., Bimbo, A., and Pala, P. 2013. Recognizing actions from depth cameras as weakly aligned multi-part bag-of poses. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 479-485).

Theodorakopoulos, I., Kastaniotis, D., Economou, G., and Fotopoulos, S. 2014. Pose-based human action recognition via sparse representation in dissimilarity space. Journal of Visual Communication and Image Representation, 25(1), pp.12-23.

Thippur, A., Burbridge, B., Kunze, L., Alberti, M., Folkesson, J., Jensfelt, P., and Hawes, N. 2015. A Comparison of Qualitative and Metric Spatial Relation Models for Scene Understanding. In AAAI (pp. 1632-1640).

Van de Weghe, N., Cohn, A., De Tre, G., and De Maeyer, P. 2006. A Qualitative Trajectory Calculus as a basis for representing moving objects in Geographical Information Systems. Control and Cybernetics, 35(1), pp. 97-119.

Vemulapalli, R., Arrate, F., and Chelappa, R. 2014. Human action recognition by representing 3D skeletons as points in a lie group. In Proceedings of the IEEE conference on computer vision and pattern recognition, (pp. 588-595).

Wang, P., Li, W., Li, C., and Hou, Y. 2016. Action Recognition Based on Joint Trajectory Maps with Convolutional Neural Networks. IEEE Transactions on Cybernetics.

Xia, L., Chen, C., and Aggarwal, J. 2012. View invariant human action recognition using histograms of 3D joints. In computer vision and pattern recognition workshops(CVPRW), 2012 IEEE Computer Society Conference on (pp. 20-27). IEEE.