Joint classification of actions and object state changes with a latent variable discriminative model

Efstathios Vafeias[1] and Subramanian Ramamoorthy[2]

***Abstract*— We present a technique to classify human actions that involve object manipulation. Our focus is to accurately distinguish between actions that are related in that the object's state changes define the essential differences. Our algorithm uses a latent variable conditional random field that allows for the modelling of spatio-temporal relationships between the human motion and the corresponding object state changes. Our approach involves a factored representation that better allows for the description of causal effects in the way human action causes object state changes. The utility of incorporating such structure in our model is that it enables more accurate classification of activities that could enable robots to reason about interaction, and to learn using a high level vocabulary that captures phenomena of interest. We present experiments involving the recognition of human actions, where we show that our factored representation achieves superior performance in comparison to alternate flat representations.**

I. Introduction

Among the many perceptual modalities available to a modern robotic system, vision is perhaps the richest in terms of the variety of information it captures about the external world and about the agents within that world. This very richness also makes interpretation highly ambiguous and often brittle. One reason why a robot might try to interpret the visual feed is to identify actions and activities being performed by agents in the environment. Given the aspirations of the robotics community to introduce robots into human environments, robots should be competent in interacting with people and recognizing human actions.

Activity recognition is typically conceptualised as a classification problem, extracting an underlying context from variability in motion, shape and other confounding factors. The focus of this paper is on a method to improve this process of extracting activity categories, by jointly analysing the actions of the human user and of the object that is the target of the activity. The goal of such a method is to enable the identification of activity categories that are behaviourally meaningful, hence useful in representing and learning of higher level interactions.

Early computer vision methods for action classification were primarily concerned with variability in motion, e.g., of body pose. While these approaches have enjoyed successes in applications that require distinctions between sequences of poses, they have often succeeded by ignoring interactions with the environment that change the context of the action. In this paper, our focus is on trying to model this notion of *context* as well. Unlike methods that are based, say, on features of single frames within the activity, we model spatio-temporal interactions with objects that define the classification outcome of an action. This makes our method suitable for the understanding of activities which are best defined by the joint evolution of the state of an object in the environment and the changes in body poses that cause that state to change.

We present a technique that learns to classify such interactions, from video data, and performs better than alternate baselines due to its use of the joint information in the recognition process. We base our work on object information, without explicitly identifying the object, showing that spatio-temporal relationships are sufficient to improve the performance of activity recognition. We believe this is better suited to the needs of incremental and lifelong learning because the notion of tracking the motion of a not-yet-fully-modelled object conceptually precedes the more sophisticated task of identifying and making inferences about the detailed properties of the object in question.

In order to explain the concept intuitively before jumping into detail, consider the fact that two actions - picking up and slightly pushing an object - can appear highly aliased and difficult to disambiguate unless one also considers what happened to the object: did it leave the hand and roll away or did it move with the hand away from the surface of a table? The disambiguating signal is neither the pose of the hand nor the identity of the object. Instead, it is the joint hand-object movement. Incorporating such structure into our models is key to learn 'symbolic' relational concepts.

In this paper, we build upon previous work on action classification based on state of the art statistical learning techniques. Given our intention to work with sequential data, we adopt a discriminative sequential classifier, Conditional Random Fields(CRF)[1]. Our model is a variation of the hidden state CRF[2], which allows us to consider the object-action spatio-temporal dependencies upon action classifications. The method is experimentally evaluated in Section IV, where we show that mutually modelling actions and object movements can significantly boost the recognition performance, when these actions involve objects.

II. Related Work

The idea that objects and actions are intertwined and highly dependant was originally proposed by the psychologist James J. Gibson, who coined the term *affordance*[3]. *Affordance* refers to a quality of an object or the environment

[1]Efstathios Vafeias is with the School of Informatics, University of Edinburgh, EH8 9QT Edinburgh, UK, email: e.vafeias@sms.ed.ac.uk

[2]Subramanian Ramamoorthy is with the School of Informatics, University of Edinburgh, EH8 9QT Edinburgh, UK, email: s.ramamoorthy@ed.ac.uk

that allows an agent to perform an action. In the field of robotics, the idea of *affordances* has been explored from various viewpoints. One popular approach is to apply known motor actions to either segment objects or learn about the affordance possibilities [4][5]. Other approaches consider the role of objects in imitation learning and use them to support motor experiences[6]. Kruger et. al.[7] introduced a framework that describes how object state changes can help to form action primitives. In other cases human demonstration is used to visually infer object affordances[8]. Furthermore, Aksoy et. al. [9] represent relations between objects to interpret human actions in the context of learning by demonstration.

There is a large corpus of work in human activity recognition in computer vision and pattern recognition illustrating its utility as a test case for sequential classifiers. Different forms of input to recognition methods include still images, video streams and video with 3D data. Approaches such as those presented in [10][11] combine pose estimation and object information for single frame inference, to learn the mutual context of actions and objects. These methods have been tested on databases where objects provide enough information, but almost no temporal dependencies are required to classify the image, e.g. holding a racket and taking the serving pose is very likely to be assigned with playing tennis. In our work, we are interested in recognizing more subtle actions that differ at a lower level, where object recognition itself is not enough to generally characterize the activity. The same action can be performed with multiple objects, thus we focus especially on temporal dependencies. Other methods[12] to classify activities from videos create visual vocabularies of features that capture spatio-temporal statistics, and then feed them into well established classifiers like SVMs or AdaBoost.

Kjellstrom et al.[8] use a Factorial CRF to simultaneously classify human actions and object affordances from videos. The difference from our work is that they use object detection that assumes known object instances and accurate hand pose segmentation. This is a valid setting for imitation learning, yet difficult to achieve in other activity recognition scenarios, especially when we do not yet have detailed object labels. While the FCRF and HCRF methods used in this paper are similar in the way they split object and action information, Kjellstrom et al.[8] consider factorization separately and predict object-action combinations, however we use a joint factorization of object and action hidden states to classify them. Additionally we use hidden states to explore structure from raw input, rather than manually annotating, sequences which contributes to make the supervision part of the algorithm less tedious.

Unlike the previous work [13][11] on classification without an explicit notion of time scale, we also want to model long term temporal relationships of the actions. In this paper, we describe a classifier that accounts for contextual information, and we show that the state of the object can greatly improve the classification of subtle actions. A key feature of many actions, is that they consist of segments of simpler homogeneous motions. Our approach exploits this observation by splitting the trajectory into smaller segments without losing valid information from the motion. Indeed, a key attribute that distinguishes our work from related prior work is that by working at a larger temporal scale, we capture structure in the activities somewhat more efficiently and in a way that is better suited to our applications of interest, such as human-robot coordinated actions.

III. METHOD

We use a hidden state CRF, which is a discriminative probabilistic model that defines a conditional distribution over sequence labels given the observations. In the remainder of this section we present our training and classification methodology.

A. The model: Hidden State CRF

Hidden state Conditional Random Fields (HCRF) are random fields that include latent variables in their structure, shown to perform well in a wide range of problems. The CRF model was first introduced by Lafferty et al.[1] and has been popular in natural language processing, although it has been increasingly gaining traction in computer vision. Quattoni[2] extended the CRF framework to incorporate hidden variables in order to capture the spatial relationships of object parts in images, while Wang et. al.[14] used a HCRF for gestural recognition.

Like any undirected graphical model, we represent variables as nodes($\mathcal{V}$) and edges($\mathcal{E}$) accounting for variable correlation, forming a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The graph is factorized and represented as a conditional probability distribution. One of the major advantages of the HCRF is that the latent variables can be dependent on arbitrary features of the observation sequence, giving them the ability to capture long-term temporal, spatial or contextual dependencies. The hidden variables at each time t have the potential to select arbitrary subsets of observations, giving them the ability to depend on observations from any frame of the sequence. Selecting subsets of the full observation set is commonly used in image processing as a way to express spatial relationships. We choose to separate object-related features and body based features connected to different latent variables at each time step t. We create nodes that depend on the f_t^{obj} object observations, and nodes that depend on f_t^{skel}, the skeleton tracking observations.

The observation sequence is represented as $X = [x_1, x_2, ..., x_T]$, and each observation x_t is an object-action pattern represented by 2 feature vectors, $f_t^{skel} \in \Re^{18}$, $f_t^{obj} \in \Re^6$. The model has two sets of latent variables, $\mathbf{s}^{obj} = [s_1^o, s_2^o, ..., s_T^o]$, and $\mathbf{s}^{skel} = [s_1^s, s_2^s, ..., s_T^s]$, each node having zero and first order dependencies. The connectivity of the model is represented in Figure 1. Each pair of nodes is assigned to a different set of hidden states according to its type. Latent variables referring to skeleton s_i^s are assigned hidden state values from the set $h_i^s \in \mathcal{H}^s$, and object variables s_i^o, are assigned from a different set $h_i^o \in \mathcal{H}^o$. The

conditional probability distribution of label class is expressed as

$$p(y|x;\theta) = \sum_{\mathbf{s}} p(y,\mathbf{s}|x;\theta) = \frac{1}{Z(y|x;\theta)} \sum_{\mathbf{s}} e^{\Psi(y,\mathbf{s},x;\theta)}$$

We denote as $\mathbf{s} = [\{s_1^o, s_1^s\}, \{s_2^o, s_2^s\}, ..., \{s_T^o, s_T^s\}]$ a chain of pair nodes(object - skeleton) and their states from their corresponding sets. The normalizing partition function is given by

$$Z(y|x;\theta) = \sum_{y \in Y, s} e^{\Psi(y,\mathbf{s},x;\theta)}$$

The model is factorized through the potential function $\Psi(y, s; \theta) \in \Re$. The potential function is parametrized by θ, and its purpose is to measure the compatibility between the observation sequence X, the hidden state configuration, and a label y. The parameter vector θ has 3 components $\theta = [\theta^v, \theta^e, \theta^y]$. Each of the three vector components is used to model a different factor of the graph. The first component θ^v models the dependencies between the raw features[1] f_t^{skel}, f_t^{obj} and the hidden states $h_i \in H^s$, $h_i \in H^o$. The length of the vector θ^v is $(d^s \times |H^s|) + (d^o \times |H^o|)$. The component θ^e models the connectivity between the hidden states, which, for a fully connected graph like the one we use, has length for $(|Y|\times|H^s|\times|H^s|)+(|Y|\times|H^o|\times|H^o|)+ 2 \times (|Y|\times|H^s|\times|H^o|)$. The θ^y vector corresponds to the links between the hidden states and the label node y, and is of $(|Y|\times|H^s|+|Y|\times|H^o|)$. We define $\Psi(y, \mathbf{s}, x; \theta)$ as a summation along the chain

$$\begin{aligned}\Psi(y,\mathbf{s},x;\theta) = &\sum_{j=1}^{T} \varphi(f_j^s, \theta^v[s_{j|s}]) + \sum_{j}^{T} \theta^y[y, s_{j|s}] \\ &+ \sum_{k=1}^{T} \varphi(f_k^o, \theta^v[s_{k|o}]) + \sum_{k=1}^{T} \theta^y[y, s_{k|o}] \\ &+ \sum_{j=2}^{T} (\sum_{k=s,o} \theta^e[y, s_{j|k}, s_{j-1|k}]) + \theta^e[y, s_{j|o}, s_{j|k}]\end{aligned}$$

In the above definition of $\Psi(y, \mathbf{s}, x; \theta)$, the function φ is the inner product of the features at time step j and the θ^v parameters of the corresponding hidden state. The $\theta^y[y, h_j]$ term stands for the weight of connection between the latent state and the class y, whereas $\theta^e[y, h_j, h']$ measures the dependency of state h_j to state h' for the given class y.

The above graphical model is an abstract way to think of learning from sequential data, and may be combined with a variety of features depending on the application. It can be expanded to work with a variety of human-object, human-environment or even human-human interactions. In this paper we will focus on a specific setup that we use in our experiments. In this setup, we wish to classify the type of activity involving a person manipulating objects in the scene.

[1]In cases were the observation window is $\omega > 0$, then $f_t = [f_{t-w}, f_{t+w}]$, so as to include all the raw observations of the time window ω.

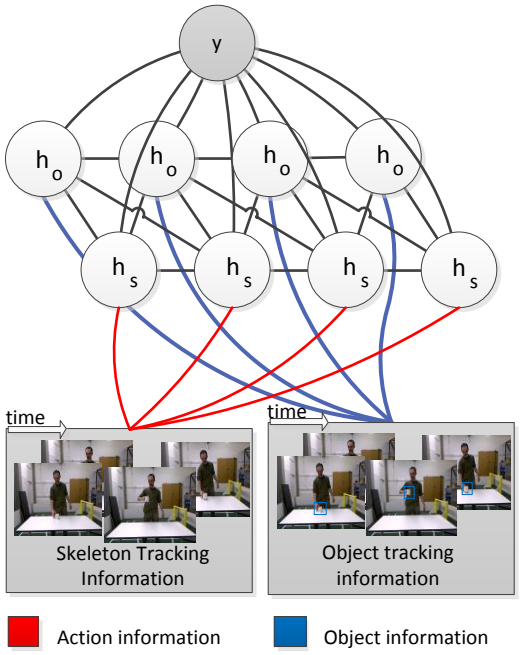

Fig. 1: A segmented sequence is shown at the bottom of the figure. White nodes represent latent variables of the model, blue lines represent object-related factors, while red lines are used to represent action-related factors. This allows our model to explicitly distinguish between action and the outcome of an action on the manipulted object. In Conditional Random Fields, the latent variable models the dependence between each state and the entire observation sequence in order to deal with the variable length of observations each state is dependant on a window of observations, $[x_{t-\omega}, x_{t+\omega}]$.

B. Implementation

At each given time step we track a set of features that correspond to the body posture of the person and the object that is being manipulated. We use the skeleton tracking[15] that is provided with the OpenNi SDK[2], the tracker provides 18 joint positions, $j_i = \{x_i, y_i, z_i\}$. The object detection routine is performed at initialization of the algorithm. Assuming that all objects are supported on a plane we try to find the largest plane that fits our point cloud. Following the plane fitting process we identify Euclidean clusters of points whose projection falls into the convex hull of the plane points. Each cluster is treated as a potential object, and only objects that fall within close distance to the user's hands are considered for tracking. For object pose tracking, we use an of-the-shelf Monte Carlo algorithm, created by Ryohei Ueda and implemented as part of the Point Cloud Library[16]. To calculate the likelihood of the poses it uses weighted metrics based on point cloud data and RGB color values from the

[2]The OpenNI framework is an open source SDK, more info: http://www.openni.org

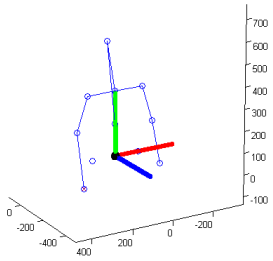

Fig. 2: Joint positions are transformed to a new coordinate system with x-axis aligned to the mid point of the hips and the left hip, y-axis defined by the mid point of hips and the shoulder center, while z-axis is the normal defined by x-y plane.

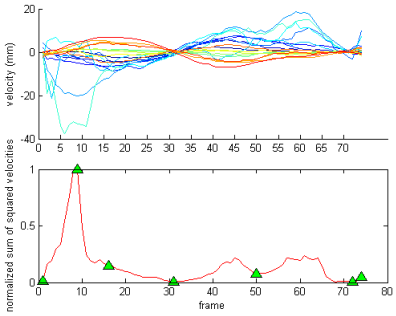

Fig. 3: Top: Joint velocities at each timestep. Bottom: Shows the normalized sum of squared velocoties. The green triangles note the split positions. *This figure is best viewed in colour.*

video stream. Object pose information at each time step t is represented as a 6D vector containing position and rotation vectors, $o_t = \{x, y, z, roll, pitch, yaw\}$.

While the combination of depth and intensity images can provide a rich set of features, our strategy is to classify the actions with a minimal set. This decision allows us to stress the importance of learning the structure of interaction between object and body motion. The temporal relationships between the state distribution of human actions and the object's spatial changes affect the classification of a sequence. We represent a sequence of length T as $X = [x_1, x_2, ..., x_T]$, and each observation at time t is composed of f_t^{skel}, f_t^{obj}, which are the features extracted from the skeleton tracking and features from object tracking respectively.

1) Pose features: To create the pose features, we use the 3D body joint locations to build a compact representation of postures. The Kinect sensor offers a real-time estimation of joint positions in the scene. To create our representation we use 6 joints, L/R shoulder, L/R elbow and L/R hand. We transform the positions to a skeleton centric coordinate system and we take the center of hips as the center of the coordinate system, Figure 2 shows the axes of the new reference frame. By choosing a frame transformation aligned to the direction the person is facing, the skeleton configuration becomes independent of the viewpoint. The Cartesian coordinates of each joint are transformed into spherical coordinates and the radius is omitted, each joint is represented by a tuple $j_t^i = (\varphi_t^i, \theta_t^i)$. The resulting viewpoint invariant feature vector has length of 6×2 = 12.

2) Hand features: While pose features capture the static joint configuration we also want to include dynamic information about the motion performed at each time step. In order to incorporate motion patterns to our feature set, we compute the joint velocities of the L/R hand.

3) Object features: The object tracker provides information about the trajectory of an object's center of mass. We use the trajectory to compute the object velocity and the distances from the head joint, L/R hand joint positions. The object information is a 6D vector consisting of the 3 distances and the object's velocity.

C. Managing trajectories

A typical trajectory has a length of 60-250 frames depending on the class. Our classification model is a dynamic template model, meaning that learned parameters are copied and reused in each step with different inputs. The label of the sequence is estimated by summing the potentials of each frame belonging into a specific class. Long sequences tend to result in error accumulation over time, which has an adverse effect on classification. To alleviate this problem, trajectories are split using a heuristic procedure that detects similar moving patterns and merge them into a single block. We take advantage of the fact that joint speed profiles consist of an accelerating motion segment, a maximum speed segment, and a decelerating segment. Our heuristic sums the squared speed of all joints and then finds the peaks and valleys in the new signal as in Fig.3, subject to some constraints, e.g. minimum thresholds for peaks and valleys and a minimum length of 5 frames for a split to be valid. The mean velocity of each joint is the new feature set for a segment. Merging similar frames into a single observation variable has the advantage of creating shorter sequences and thus shorter latent variable chains. While this particular trajectory segmentation method is a simple heuristic, it does manage to capture motion changes and to segment homogenous parts of the motion. Thus, it exploits the full power of our model which relies on capturing temporal correlations between varying states. Alternate trajectory segmentation methods could be used as drop in replacements without altering our overall arguments.

D. Inference and Learning of Parameters θ^*

Given a set of parameter values θ^* and a new sequence of observations X, the label y^* can be computed as

$$y^* = argmax_{y \in Y} P(y|x, \theta^*)$$

Learning in conditional random fields is treated as an optimization problem, where a θ is estimated through the maximization of the objective function(Eq. 1). The likelihood term of the objective function, $logP(y_i|x_i; \theta)$, is calculated by loopy belief propagation.

$$L(\theta) = \sum_{i=1}^{N} logP(y_i|x_i;\theta) - \frac{||\theta||^2}{2\sigma^2} \quad (1)$$

$$\theta^* = argmax_\theta L(\theta) \quad (2)$$

where N is the total number of training sequences in the dataset $D = \{x_i, y_i\}, i = 1, ..., N$, and θ are target parameters. The second term, $-\frac{||\theta||^2}{2\sigma^2}$, is the L_2 regularization penalty that we use to avoid overfitting. In a simple CRF, with no hidden states, the likelihood function $L(\theta)$ is convex taking the form of an exponential of the potential function. However, in the case of hidden state CRF, we need to marginalize over the hidden states and thus create a summation of exponentials which makes our objective function non-convex. To optimize this function we use gradient ascent with various starting points to avoid local maxima. The optimization algorithm we chose is the L-BFGS[17] which is shown to perform well with a large number of parameters.

IV. Experimental Results

To evaluate how our system deals with actions that result in object state changes, we have created a new dataset[3] of people using various objects. Our dataset consists of actions with substantial motion similarities, making it hard to naively distinguish between them without knowing the effect on the objects of the environment. Our baseline comparison is against two different implementations of the HCRF[14]. While HMM models are ubiquitous tool for modelling sequential data, applying them on high-dimensional real-valued observations is not a trivial task and all freely available methods, that we know of, do not deal with high dimensionality. For this reason, we are unable to compare against alternatives such as HMM variants. We report on experiments with the following models:

- B model: a simple HCRF model with a single chain of latent variables trained on action features only.
- B-O model: a HCRF model with a single chain of latent variables and trained on action and object features that are modelled through a single observation variable.
- Full Model: Our model as explained in section III.

These models have been selected to bring out the importance of object interactions in activity and behaviour understanding. HCRF models perform reasonably in classifying sequential data, however we show that altering the basic model to explicitly consider the interaction boosts the overall recognition performance.

A. Dataset

Our overall goal is to model human activity that involves object manipulation. In order to do that we had to create a dataset that is suited to our needs. We recorded 918 sequences from 12 different people. We selected 5 different actions that have similar appearance and statistics if seen out of context (i.e., without the object information). The action set we recorded is based on the categories $\mathcal{A}$={drink, push, pickup, stack objects, read}, as in Fig. 4. The actions were not thoroughly scripted, as each person was given simple oral instructions about each action and was not given an explicit instruction regarding preferred or typical trajectories to follow. This is how we expect regular people to behave in everyday base. On each repetition, users freely choose how to perform the action, which hand to use, what the starting position of the object should be, speed of execution, etc. Most subjects did not repeat the same motion, so the majority of the recorded sequences have a wide range of motion variations. The actions were recorded with a Kinect camera at a rate of 30Hz, and the time length of each sequence were from 50 frames to 250 frames depending on the action class.

Fig. 4: Images from the action sequences, from top to bottom, images show action: drink, push, stack, read

All the sequences were recorded in our lab in a fairly generic setting (Fig. 4). Users stood in front of a table on which the objects of interest were placed, at distance of $2 \sim 2.5$ meters away from the Kinect camera. The difficulties in recognizing motions in the dataset are primarily related to motion similarity and occlusions. Occlusions seriously affect the performance of the skeleton tracking algorithm, which is really designed to work best in an occlusion free environment. In order to strengthen our hypothesis and test our model, we chose highly similar (i.e., aliased) motions to be part of the dataset. For example, reaching to pick an object produces a similar body posture sequence as pushing an object on the table. Similarity in motion can be found in picking and stacking objects, the reaching part of the motion and the withdrawing of the person are very similar.

B. Models and implementation details

The full model is expected to learn the spatio-temporal action - object state transitions and be able to outperform its simple counterpart where no information fusion is performed. To optimize performance, we search for parameter configurations, keeping the one with the best score on the cross validation set. The free model parameters that need to be determined are the number of hidden states in the sets, $\mathcal{H}^s, \mathcal{H}^o$, the observation window length, the standard deviation (σ) of the L_2 regularization factor and the starting values of θ for the gradient search. For hidden states, we experimented with a number of different state sets, varying from 3 to 8 for the object latent variables and from 4 to 12 for the latent variables that depend on skeleton nodes. Based on the average sequence length we experimented with window sizes of $\omega = 0, 1, 2$. After determining the best pair of hidden state numbers, we set them to a constant value and

[3]Our dataset will be made available at: http://wcms.inf.ed.ac.uk/ipab/autonomy/

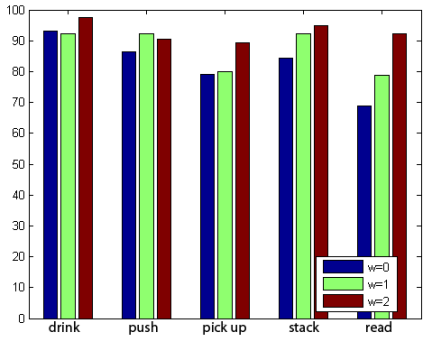

Fig. 5: Full model with different observation windows $\omega = 0, 1, 2$, reporting the F_1 score for each class. Average F_1 scores for $\omega = 0, 1, 2$ are 81.35%, 87.17% and 92.96% respectively.

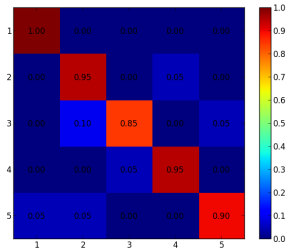

Fig. 6: Confusion matrix of our test results with average F_1 score 92.96%. Parameters: $h^o = 4, h^s = 7, \omega = 2$.

then tuned the L_2 standard deviation parameter. The σ of the regularization term was set to $\sigma = 1^k$ where k=-3,...,3.

To investigate how information fusion affects the classification performance we implemented two alternative hidden state CRF models as comparison methods. Both models contain a single layer of latent variables, meaning we use one latent variable per time-step to form single a chain for each sequence. The first model is trained only with body motion information (noted as B model), the object features are neglected, while in the second HCRF (noted as B-O model) the object features and body motion are modelled through a single observation variable X_T. Our aim is to show that we can gain accuracy compared to a model that doesn't consider object context, but also showing that modelling the action and the object information through different observation and latent variables can further improve the performance of the model. Both models have the same free parameters, number of hidden states, regularization factor and the length of the observation window. Parameters are tuned via the same grid search technique as mentioned before.

C. Results

To evaluate the performance of the different models we report the F_1 score which is a measure of accuracy that considers both the precision and recall. The F_1 score is defined as $F_1 = 2 \times \frac{precicion \times recall}{precicion + recall}$. In figure 5, we summarize the F_1 score of our approach for each class with 3 different observation window lengths($\omega = 0, 1, 2$). For each window parameter we report the best configuration of hidden states and regularization term, which differs for each model. The bar graph shows performance is correlated to the temporal relationships between the hidden states and the observations. Figure 6 shows the confusion matrix of our best results on this dataset with average F_1 score for all the classes of 92.96%. From the confusion matrix, we can see that the lowest classification rate corresponds to the pick class. Picking is a sub-event occurring in every action of the dataset, so in noisy sequences or sequences which have been mistreated during the trajectory splitting, this can cause misclassification. Intuitively when we consider spatio-temporal process we think of them in terms of past state, present state, and future state and the change of state, encoding the past-present-future information in each hidden state and not only through latent state transitions creates a more powerful representation of the action.

The full model appears to perform well on our current dataset, but in order to show the importance of object information, it is crucial to compare it to similar models that discard the object information or do not implicitly model it. In Figure 7 we show the performance of the three models in each class and in Figure 8 we report the mean F_1 score of each model with the corresponding standard deviation. Between the simple B model and the full model, there is a significant increase in performance by 13.84%. In Fig. 7, we see that B-Single performs better than our model on the drinking action and matching our models performance on the stacking action. The drinking action is particularly distinctive, so that even the simpler baseline method is already able to capture it and our method provides no added advantage here. The case of the stacking action is more interesting in that the real reason for B-single doing well is a favourable class bias. The stacking action has a similar profile to the pickup action for the whole length of the sequence, while at the start of the sequence it shares profiles with read and push. This class bias pushes the score for stack to equalize the performance of our model. Adding the object information to the training set for the B-O model increases the average F_1 score from 79.11% to 88.83%, and overall there is a lower variance in the accuracy between classes. Comparing the B-O model with our full implementation, we observe a notable increase in the average F_1 score from 88.83% to 92.96% while halving the standard deviation between class accuracy.

V. Conclusions

The main contribution of our paper is a novel method for activity modelling. Specifically, we present an algorithm that improves upon the state of the art in recognition of actions, which is a key ingredient of HRI where a robot needs to understand the goals and context of human action based on overt behaviour as seen from changes in body pose. Our

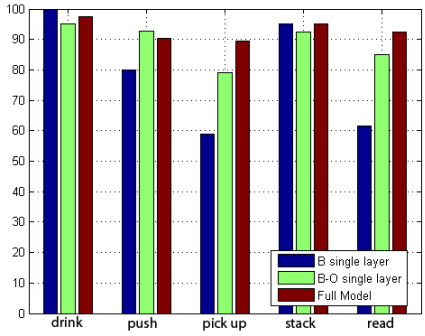

Fig. 7: F_1 scores of each class for different models. *B single layer:* Only body motion features are trained. *B-O single layer:* Both body motion and object features are modeled under the same observation variable. *Full Model:*The full model as presented in Section III.

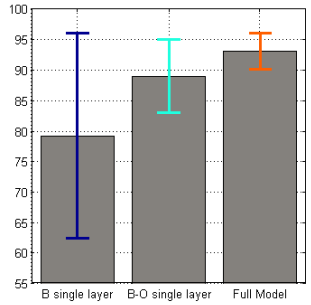

Fig. 8: F_1 scores for each model, class mean and its standard deviation. Our full model achieves the best result while maintaining a very low variance between class accuracy. Our implementation shows a more robust approach on how one can jointly classify actions that result to object state changes.

main observation is that categorization is vastly improved when we jointly model the changes in body pose and the state of the object that is being acted upon, i.e., the effects on the environment of the person's movement. We do this in the setting of a state of the art statistical learning algorithm for discriminative classification, the HCRF. Our experiments demonstrate that thinking of actions in terms of motion and outcome yields significant overall improvement. We view this work as a first step in understanding how to devise the ability to decode human activity at finer levels, which lead to improved human - robot interactions and learning by demonstration capabilities. Our future plan is to continue this investigation to understand how different time scales, the much longer as well as the subtle but shorter, can be captured in a similarly factored way.

ACKNOWLEDGMENT

We thank Aris Valtazanos for helpful discussion and assistance with aspects of presentation. This work has taken place in the Robust Autonomy and Decisions (RAD) group, School of Informatics, University of Edinburgh. The RAD Group is supported in part by grants from the UK Engineering and Physical Sciences Research Council (EP/H012338/1), the European Commission (TOMSY Grant 270436, FP7-ICT-2009.2.1 Call 6) and a Royal Academy of Engineering *Ingenious* grant.

REFERENCES

[1] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," 2001.

[2] A. Quattoni, S. Wang, L.-P. Morency, M. Collins, and T. Darrell, "Hidden conditional random fields," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 10, pp. 1848–1852, 2007.

[3] J. J. Gibson, "The ecological approach to the visual perception of pictures," *Leonardo*, vol. 11, no. 3, pp. 227–235, 1978.

[4] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, and G. Sandini, "Learning about objects through action-initial steps towards artificial cognition," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 3. IEEE, 2003, pp. 3140–3145.

[5] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, "Learning object affordances: From sensory–motor coordination to imitation," *Robotics, IEEE Transactions on*, vol. 24, no. 1, pp. 15–26, Feb.

[6] M. Lopes and J. Santos-Victor, "Visual learning by imitation with motor representations," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 35, no. 3, pp. 438–449, June.

[7] V. Kruger, D. Herzog, S. Baby, A. Ude, and D. Kragic, "Learning actions from observations," *Robotics & Automation Magazine, IEEE*, vol. 17, no. 2, pp. 30–43, 2010.

[8] H. Kjellström, J. Romero, and D. Kragić, "Visual object-action recognition: Inferring object affordances from human demonstration," *Computer Vision and Image Understanding*, vol. 115, no. 1, pp. 81–90, 2011.

[9] E. E. Aksoy, A. Abramov, J. Dörr, K. Ning, B. Dellen, and F. Wörgötter, "Learning the semantics of object–action relations by observation," *The International Journal of Robotics Research*, vol. 30, no. 10, pp. 1229–1249, 2011.

[10] W. Yang, Y. Wang, and G. Mori, "Recognizing human actions from still images with latent poses," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2030–2037.

[11] B. Yao and L. Fei-Fei, "Modeling mutual context of object and human pose in human-object interaction activities," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 17–24.

[12] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos in the wild," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1996–2003.

[13] A. Gupta, A. Kembhavi, and L. Davis, "Observing human-object interactions: Using spatial and functional compatibility for recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 10, pp. 1775–1789, Oct.

[14] S. B. Wang, A. Quattoni, L.-P. Morency, D. Demirdjian, and T. Darrell, "Hidden conditional random fields for gesture recognition," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2. IEEE, 2006, pp. 1521–1527.

[15] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Commun. ACM*, vol. 56, no. 1, pp. 116–124, Jan. 2013. [Online]. Available: http://doi.acm.org/10.1145/2398356.2398381

[16] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[17] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization," *ACM Trans. Math. Softw.*, vol. 23, no. 4, pp. 550–560, Dec. 1997. [Online]. Available: http://doi.acm.org/10.1145/279232.279236