Human activity recognition by combining discriminative and generative classifiers

Ioannis Kaloskampis
Cardiff University, UK
kaloskampisi@cardiff.ac.uk

Yulia Hicks
Cardiff University, UK
hicksya@cardiff.ac.uk

***Abstract*—In this article, we propose a novel algorithm for the recognition of complex activities in multimedia streams. The algorithm consists of a discriminative feature classifier based on random forests and a generative classifier, for which we use the hierarchical hidden Markov model. The discriminative feature classifier checks the existence or absence of the steps required for the execution of an activity, while the generative classifier encodes the ordering of these steps. The parameters of the classifier are learned automatically from expert labelled data. The classification output is a label indicating, for an input multimedia stream illustrating a complex activity, the type of the activity performed in the stream and whether this activity was performed in a correct or incorrect/anomalous manner.**

Results for the publicly available bridge design dataset show that our algorithm offers higher accuracy in activity recognition than other leading methods.

***Keywords*—activity recognition, random forest, hierarchical hidden Markov model.**

I. Introduction

The recognition of human activities in multimedia streams is an important research topic, with many applications in the fields of surveillance, security and digital media. Most of the work previously carried out in this area focuses on short, everyday tasks with relatively small action vocabulary and activities that can be performed in a limited number of ways. In this article, we propose a new algorithm for the recognition of long activities in tasks which can be executed in a variety of ways. Our algorithm can also detect erroneous executions in such tasks, which can be considered as anomalies.

An activity is typically represented as a sequence of its constituent actions [1]. There are currently five approaches to activity modelling:

Grammar-driven representations, such as context-free grammars [2] and stochastic context-free grammars [3] which model activities with sets of production rules that describe all possible executions. These methods cannot be applied when the structure of the activity is not known *a priori*.

Vector space models, *e.g.* [4], where an activity is represented as a vector of its constituent actions. The drawback of the approaches of this category is that they lack a facility to model temporal dependencies between actions.

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant number EP/K014307/1 and the MOD University Defence Research Collaboration in Signal Processing.

Local event statistic methods, which capture neighbouring temporal relations between an activity's constituent actions, such as n-grams [1], suffix trees [5] and augmented bags-of-words [6]. In noisy datasets these neighbouring relations sometimes become less characteristic of the performed activity.

Statistical graphical models, such as the hidden Markov model (HMM) [7] and the variable length Markov model [8]. Although they are capable of encoding the temporal dependencies between actions efficiently, they cannot represent the natural hierarchical structure of complex activities [9].

Extensions of statistical graphical models in a hierarchical manner, *e.g.* the hierarchical hidden Markov model (HHMM) [10], [9] and the layered hidden Markov model [11]. These models can capture complex hierarchy but sometimes prove to be sensitive to noise.

In this paper, we propose a new algorithm for activity recognition which addresses the shortcomings of previous approaches. It can model activities whose exact structure is not previously known. It is capable of efficiently representing the natural hierarchy of complex activities and encode the temporal relations between their constituent actions. Our algorithm combines a discriminative feature classifier based on random forests (RF) [12] and a generative classifier for temporal analysis, for which we use a HHMM. The discriminative feature facility checks the existence or absence of the steps required for the execution of an activity, while the generative model encodes the ordering of these steps. Our algorithm can be applied as it stands to any task which involves complex activities, as all of its components are learned automatically from training data.

The proposed algorithm can be used to detect erroneous or anomalous activity executions. When such executions are present in the training dataset, this is achieved by building separate sub-models corresponding to erroneous executions. In the absence of such data, our algorithm can detect anomalies by assessing the confidence scores assigned to various activity executions during the classification process.

We evaluate our algorithm's performance for the bridge design dataset [13] and compare it with several state of the art algorithms. The results show that our algorithm offers higher accuracy in activity recognition than other leading methods.

The remainder of the paper is structured as follows. Section II describes the proposed activity recognition algorithm. Experimental results are presented in Section III; the paper is concluded in Section IV.

II. Combining discriminative and generative classifiers

In this section we present a methodology to analyse activities using a classifier which combines the discriminative capabilities of RFs and the efficiency of HHMMs in encoding complex temporal relations between an activity's constituent actions. Classification is performed in a supervised manner: a model is first learned automatically using a set of labelled sequences; this model is then used to analyse novel input sequences (*i.e.* sequences not present in the training dataset).

We assume that the action sequences illustrating activities are pre-extracted from the multimedia streams. This can be achieved, for example, as described in [13], [3].

A. Model training

Consider a set U_{tr} of N_{tr} labelled action sequences, $U_{tr} = \{U_{tr,i} = (S_i, c_i)\}$ where $S_i \in \mathcal{S}$ is an action sequence and $c_i \in \mathcal{C}$ is the sequence's class label. The set U_{tr} will be used to train the model. The classifier comprises two parts, a RF classifier and a HHMM classifier which are trained separately.

1) RF Classifier Training (discriminative classifier): An ensemble RF classifier comprising N independent decision trees is built. The n^{th} tree of the classifier is denoted by $f_n(S_i, \theta_n) : \mathcal{S} \rightarrow \mathcal{C}$ mapping each element of the sample space $\mathcal{S}$ to a label in the label space, $\mathcal{C}$. θ_n is a random vector containing the tree's stochastic elements (*e.g.* the randomly subsampled training set or selected random tests at the tree's decision nodes). We denote the entire forest by $\mathcal{F} = \{f_1, ..., f_N\}$.

Each tree is grown as follows: a bootstrapped sample of the training dataset, $U_\beta = \{U_{\beta,j} = (S_j, c_j)\}$ is taken for the tree. For each non-leaf node of the tree a split function has to be defined, which provides the optimal separation of the training sequences:

$$f_\Phi(U_{\beta,j}) \in \{0, 1\} \tag{1}$$

where Φ is the set of parameters of the split function. The function evaluates one or more features of sequence $U_{\beta,i}$ and decides about whether it will be sent to the left $(f_\Phi(U_{\beta,j}) = 0)$ or right child $(f_\Phi(U_{\beta,j}) = 1)$ of the node. The parameters Φ are optimised during the training process, which is summarised in the following paragraphs.

The algorithm starts at the root node of the tree with the training set $U_\beta = U_{node}$. A random set of parameters, $\Phi = \{\phi_k\}$ is generated and the training set U_{node} is divided into two subsets, U_L and U_R, $\forall \phi \in \Phi$ as follows:

$$U_L(\phi) = \{U_{\beta,j} \in U_{node} | f_\phi(U_{\beta,j}) = 0\} \tag{2}$$

$$U_R(\phi) = \{U_{\beta,j} \in U_{node} | f_\phi(U_{\beta,j}) = 1\} \tag{3}$$

The split parameters ϕ^* are selected so that they optimise a gain function g:

$$\phi^* = \arg\max_{\phi \in \Phi} g(\phi, U_{node}) \tag{4}$$

where the gain function g is given in the equation:

$$g(\phi, U_{node}) = H(U_{node}) - \sum_{M \in \{L,R\}} \frac{|U_M(\phi)|}{|U_{node}|} H(U_M(\phi)). \tag{5}$$

The function H measures the gain of the classification accuracy of the children nodes in comparison to the current node. The following entropy-based classification function H is proposed in [14]:

$$H(U_{node}) = -\sum_c p(c|U_{node}) log(p(c|U_{node})) \tag{6}$$

where the class probability, $p(c|U_{node})$, can be calculated from the equation

$$p(c|U_{node}) = \frac{|U_c^{node}| \cdot r_c}{\sum_c (|U_c^{node}| \cdot r_c)} \tag{7}$$

with U_c^{node} the set of sequences with class label c reaching the studied node after training and

$$r_c = \frac{|U_{tr}|}{|U_c|} \tag{8}$$

where U_c the set of sequences with class label c within the whole training set, U_{tr}.

If the stopping criteria are not satisfied, the tree continues to grow using the subsets U_L and U_R. Else a leaf node is created which stores the statistics of the training data U_{node}. The class probability $p(c|L)$ at leaf L can be estimated with the equation:

$$p(c|L) = \frac{|U_c^L| \cdot r_c}{\sum_c (|U_c^L| \cdot r_c)}. \tag{9}$$

2) Feature selection: One of the advantages of using RF is that they integrate a variable importance facility which assesses the significance of each feature participating in the classification process [12]. We evaluate the proposed algorithm both with and without taking into account this facility.

To assess the importance of the variables, the algorithm predicts for every tree in the forest the classification of the out-of-bag (OOB) samples of the training set and estimates the misclassification rate. The OOB_n samples for a tree n are defined as the training samples not used during the construction of n. The misclassification rate is defined as the tree's out-of-bag error.

The values of every variable in the tree are permuted and the out-of-bag error is estimated. By comparing this error to the

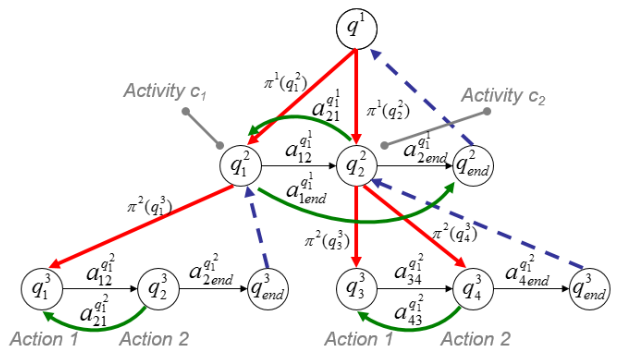

Fig. 1: The generic three-level HHMM topology used by our algorithm to encode temporal dependencies between an activity's constituent actions. Two sample activities are represented, c_1 and c_2.

misclassification rate of the tree an indication of the variable's importance is obtained. The increase of misclassification rate is defined as the variable's importance measure for the tree.

Out-of-bag errors and importance measures from all trees in the forest are then aggregated to obtain the overall out-of-bag error rate and variable importance measures.

3) HHMM Training (generative classifier): Having selected the important features using the Variable Importance assessment method, non-important features are removed from the original training sequences. Therefore, a filtered training dataset U'_{tr} of N_{tr} labelled action sequences is obtained. The filtered dataset is used to train the HHMM. When working without the variable importance facility, we train the HHMM with the unfiltered dataset, U_{tr}.

We briefly present the HHMM, following [10]. The task vocabulary, containing all possible actions in the studied task is denoted by Σ. An activity can be represented as a sequence $S = \{s_1, s_2 \dots s_T\} = s_{1:T}$ where T the length of the sequence. Each state of the HHMM is denoted by q_i^d, $d \in \{1, \dots, D\}$ with i the state index and d the hierarchy index (for the root it is $d = 1$, for production states $d = D$). The number of substates of an internal state q_i^d is denoted by $|q_i^d|$. The state index will be omitted in cases where it is clear from the context and thus a state at level d will be denoted by q^d. Each level, excluding the root level has an ending state, q_{end}^d. The state transition probabilities between the internal states at level $d+1$ are given by the matrix $A^{q^d} = (a_{ij}^{q^d})$ with $a_{ij}^{q^d} = P(q_j^{d+1}|q_i^{d+1})$ the probability of transitioning from state i to j within level $d+1$. The initial distribution over the substates of q^d is given by the vector $\Pi^{q^d} = \{\pi^{q^d}(q_i^{d+1})\} = \{P(q_i^{d+1}|q^d)\}$. Note that $P(q_i^{d+1}|q^d)$ is the probability that parent state q^d will initially activate substate q_i^{d+1}. The production states, q^D emit actions as specified by their output probability vector $B^{q^D} = \{b^{q^D}(k)\}$. In this case, $b^{q^D}(k) = P(\sigma_k|q^D)$ is the probability that q^D will produce action $\sigma_k \in \Sigma$. The set of parameters for the HHMM is denoted by $\lambda = \{\lambda^{q^d}\}_{d\in\{1,\dots,D\}} = \{\{A^{q^d}\}_{d\in\{1,\dots,D-1\}}, \{\Pi^{q^d}\}_{d\in\{1,\dots,D-1\}}, \{B^{q^D}\}\}$.

Learning the HHMM from the training dataset U'_{tr} of N_{tr} labelled action sequences is now discussed. Previous approaches using HHMM for activity recognition (*e.g.* [9]) have used a manually specified model topology. Here, the topology is automatically learned from training data, building on the generic three-level hierarchy of Fig. 1. Specifically, the first level of the HHMM is the root. The second level represents activities and consists of a number of nodes equal to the number of different activities performed in the studied dataset. The third level represents actions and comprises, for each node of the second level, a number of nodes equal to the number of different actions in the dataset. The parameters of the model are then estimated by discovering the most probable set of parameters, λ^* with $\lambda^* = \underset{\lambda}{\arg\max}\, P(\{S_t\}|\lambda)$. This is achieved with the generalised Baum-Welch algorithm [10]. The only information that the algorithm requires is the training dataset U'_{tr} of N_{tr} labelled action sequences. For more details about the Baum-Welch algorithm please see [10].

B. Activity recognition and anomaly detection

Consider the task of classifying a test dataset, $U_{te} = \{U_{te,i} = (S_i, c_i)\}$ of N_{te} labelled action sequences not included in the training dataset U_{tr}. The process is as follows:

1) Discriminative classification (RF): Non-important elements detected during the training process are removed from the test dataset and thus the filtered test dataset, U'_{te} is obtained. When working without the variable importance facility, we simply use the unfiltered dataset, U_{te}.

The filtered test dataset U'_{te} is passed to the RF classifier. For the n^{th} tree tree of the forest, each sequence S_i of the dataset will fall into a leaf. We can calculate the probability for predicting class c for a sample with the equation

$$p(c|S_i) = \frac{1}{N}\sum_{n=1}^{N} p_n(c|S_i), \qquad (10)$$

in which $p_n(c|S_i)$ is the estimated density of class labels of the leaf of the n^{th} tree where S_i falls with $p_n(c|S_i) = p_n(c|L)$. The class probability at leaf L, $p_n(c|L)$ can be directly estimated from *Eqn. 9*. Note that, as illustrated by its derivation *Eqn. 10* is obtained in a data-driven fashion; it does not require knowledge of the forms of underlying probability distributions. Therefore its classification accuracy (as reflected in the results) depends on the representativeness of the data used for training, which in our case is the filtered training dataset U'_{tr} consisting of N_{tr} labelled action sequences. The forest's multi-class decision function is given by the equation:

$$C(S_i) = \underset{c\in\mathcal{C}}{\arg\max}\, p(c|S_i). \qquad (11)$$

Assuming that the training dataset includes erroneous activity executions, classification using *Eqn. 11* results in mapping the elements of U'_{te} to classes corresponding to correctly executed activities $U'_{te,corr}$ and erroneously executed activities $U'_{te,err}$. If the training dataset does not contain erroneous executions, the classes corresponding to erroneously executed activities are not present. In this case, anomalies can be detected by assessing the confidence scores assigned to $U'_{te,corr}$ during the classification process.

	HMM			SVM			RF			RF+HHMM		
	Corr.	Anom.	Avg.	Corr.	Anom.	Avg.	Corr.	Anom.	Avg.	Corr.	Anom.	Avg.
No feature selection	72	61	67	**86**	58	72	81	78	80	81	**89**	**85**
SVM Variable Importance	72	61	67	**86**	58	72	81	78	80	-	-	-
RF Variable Importance	78	61	70	83	64	74	81	81	81	81	**89**	**85**

TABLE I: Classification results for the bridge dataset in terms of percent classification accuracy.

2) Generative classification (HHMM): Sequences of sets $U'_{te,corr}$ are passed to the HHMM. Inference is performed with the generalised Viterbi algorithm [10]. For more details about this algorithm please see [10]. It requires only the set $U'_{te,corr}$ and assigns a class to each sample of the sets.

3) Activity characterisation: The class assigned to each of these sequences of set $U'_{te,corr}$ by the HHMM is the output of the combined RF+HHMM algorithm.

The sequences of the set $U'_{te,err}$ retain the class label assigned to them by the RF classifier.

We consider as anomalies all sequences classified into classes corresponding to erroneously executed activities by either the discriminative or the generative classifier.

III. Experimental results

The framework is assessed on the 'bridge design' dataset [13]. The dataset consists of 136 sequences of length 5-15 minutes each, in which humans execute one of three complex tasks: evaluate soil condition, estimate transient loads and evaluate bridge cost. We use the partition of the dataset into testing and training data recommended in [13].

We apply our combined generative and discriminative model with and without feature selection. To select important features we apply the random forest variable importance (RFVI) facility, described in Section II-A2, on the training dataset. We repeat the process 10 times and estimate the average importance for all features. We consider features with negative importance as redundant and remove them.

The results are shown on Table I. The proposed algorithm, RF+HHMM which combines a discriminative and a generative classifier achieves the highest accuracy in anomaly detection and on average, significantly outperforming the performance of the baseline classifiers reported in [13]. The application of RFVI feature selection algorithm does not increase the classification accuracy of our algorithm. This could be attributed to the fact that the impact of unimportant features is mitigated in the first stage of the proposed algorithm (discriminative classification with RF).

IV. Conclusion

We have presented an algorithm for activity recognition and anomaly detection of complex activities in multimedia streams. The algorithm consists of a discriminative feature facility based on random forests and a generative model, for which we use the hierarchical hidden Markov model. The discriminative feature facility checks the existence or absence of the steps required for the execution of an activity, while the generative model encodes the ordering of these steps.

Results for the publicly available bridge design dataset show that our approach offers higher accuracy in activity recognition than the other leading methods.

References

[1] R. Hamid, A. Johnson, S. Batta, A. Bobick, C. Isbell, and G. Coleman, "Detection and explanation of anomalous activities: Representing activities as bags of event n-Grams," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2005, pp. 1031–1038 vol. 1. 1

[2] M. S. Ryoo and J. K. Aggarwal, "Recognition of composite human activities through context-free grammar based representation," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, 2006, pp. 1709–1718. 1

[3] H. Kuehne, J. Gall, and T. Serre, "An end-to-end generative framework for video segmentation and recognition," in *Proc. IEEE Winter Applications of Computer Vision Conference (WACV 16)*, Lake Placid, Mar 2016. 1, 2

[4] C. Stauffer and W. E. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, Aug. 2000. 1

[5] R. Hamid, S. Maddi, A. Bobick, and I. Essa, "Structure from statistics - unsupervised activity analysis using suffix trees," in *Proceedings of the IEEE Conference on Computer Vision (ICCV)*, vol. 1, Los Alamitos, CA, USA, 2007, pp. 1–8. 1

[6] V. Bettadapura, G. Schindler, T. Ploetz, and I. Essa, "Augmenting bag-of-words: Data-driven discovery of temporal and structural information for activity recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, June 2013. 1

[7] L. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989. 1

[8] A. Galata, N. Johnson, and D. Hogg, "Learning variable length Markov models of behaviour," *Computer Vision and Image Understanding*, vol. 81, pp. 398—413, 2001. 1

[9] N. T. Nguyen, D. Q. Phung, S. Venkatesh, and H. Bui, "Learning and detecting activities from movement trajectories using the Hierarchical Hidden Markov Model," *In Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 955—960, 2005. 1, 3

[10] S. Fine, Y. Singer, and N. Tishby, "The Hierarchical Hidden Markov Model: Analysis and Applications," *Machine Learning*, vol. 32, no. 1, pp. 41–62, 1998. 1, 3, 4

[11] N. Oliver, E. Horvitz, and A. Garg, "Layered representations for human activity recognition," in *Proceedings of the IEEE 4th International Conference on Multimodal Interfaces*, 2002, pp. 3–8. 1

[12] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. 1, 2

[13] I. Kaloskampis, Y. Hicks, and D. Marshall, "Complex activity recognition and anomaly detection in multimedia streams," in *10th International IMA Conference on Mathematics in Signal Processing*, December 2014, pp. 1–4. 1, 2, 4

[14] J. Gall, N. Razavi, and L. Van Gool, "An introduction to random forests for multi-class object detection," *Lecture Notes in Computer Science*, vol. 7474, pp. 243–263, 2012. 2