Concurrent Activity Recognition For Clinical Work

Afsaneh Doryab and Julian Togelius

***Abstract*— We present an approach to learning to recognize concurrent activities based on multiple data streams. One example is recognition of concurrent activities in hospital operating rooms based on multiple wearable and embedded sensors. This problem differs from standard time series classification in that there is no natural single target dimension, as multiple activities are performed at the same time. Hence, most existing approaches fail. The key innovations that allow us to tackle this problem is (1) learning to recognize base activities from raw sensor data, (2) creating artificial joint activities from base activities using frequent pattern mining and (3) handling temporal dependency using virtual evidence boosting.**

I. Introduction

Recent technological development in computing artifacts allow people to obtain and interact with information in a more social and situated manner by moving computation beyond personal devices and into publicly available devices and displays. Key technologies include portable computers, large public displays and wireless network and sensing technologies. With these technologies it is also possible to support collaborative work among people in shared physical locations. Pervasive or ubiquitous technologies computing research is concerned with technologies that allow people to work in a more flexible way, having supporting tools available anywhere rather than being tied to a specific desktop device or location. This capability requires systems being aware of their context, i.e., the physical and social situation in which they are embedded.

The context of a system may change as the system is moved to a new execution environment (mobile device) or the physical context of an embedded system changes because, for example, new people and devices enter a room [1]. As activities occurring in the physical environment play a central role in understanding the situation, there is an increasing need and intention to detect and recognize such activities using different sensing technologies, such as vision-based or embedded and body worn sensors. Recognition of daily routines and basic physical actions such as running, walking, climbing stairs, etc. have been widely studied in recent years. These studies have mostly focused on individual users in an environment such as home [2, 3] or outdoors [4, 5].

Although recognition techniques addressing individual user's activity on e.g., mobile devices have got much attention and obtained promising results, the problem of multi-user activities in shared physical environments has remained challenging and rarely studied. As most real world settings involve several people, recognition of multi-user activities in the same context becomes necessary and worth being investigated. Examples of such situations are collaborative work environments, where the activities and tasks are distributed and shared among several people, e.g., when two people repair different parts of a car or device. Each part of the task is done by one or more participants working concurrently towards the same goal. In order to build an awareness of the situation, a system should recognize activities being performed by different participants in the same situation.

The authors are with the IT University of Copenhagen, 2300 Copenhagen, Denmark. (email: {adoryab, juto}@itu.dk)

Fig. 1. Medical information relevant to a surgical operation is shown on wall displays in an operating room. The information provided is adapted based on clinical activities during the operation.

The particular focus of this research is on clinical work and finding context-aware solutions to help clinicians manage access to the extensive amount of data as an integrated part of their tasks. By taking context into consideration, more relevant information are presented to clinicians on public displays in different situations. For example, by using our approach inside an operating room (OR), the surgical activities of clinicians are recognized from the sensor inputs. These activities are then used as context to which the wall display in the OR adapts. The adaptation is in form of showing medical resources that are relevant to that ongoing operation (see e.g., figure 1).

II. Activity recognition

Activity recognition can be defined as a sequential classification problem where at each time step, the state of the activity, which depends on previous and future states, is predicted. It can be solved by structured prediction methods which combine the ability of graphical methods to model multivariate data with the ability of classification techniques to perform prediction using large sets of input features [6] . In other words, we wish to predict a vector $y = \{y_0, y_1, ..., y_T\}$ of random variables given an observed feature vector x, where each variable y_t is the activity at time t, and the input

x is divided into $\{x_0, x_1, x_T\}$. Each x_t contains a set of different feature attributes, e.g., location and identity. The traditional classification approach to maximize the number of y_ts that are correctly labeled is to learn an independent per-position classifier that maps $x \rightarrow y_t$ for each t. The difficulty, however, is that the output variables or activities often have dependencies. For example, for dish washing, one should first put the dishes in the dish washer before starting the machine.

A natural way to represent the manner in which output variables depend on each other is provided by graphical models. Two of the most popular structured models for sequential classification are Hidden Markov Models (HMMs) [7] which have long been applied to the activity recognition problem and Conditional Random Fields (CRF) [8]. The following describes the background of these two frameworks as well as their strengths and weaknesses in addressing the activity recognition problem. We discuss the reasons why CRF framework is a more appropriate choice to address the problem of sequential multiple activity recognition.

A. Hidden Markov Models

HMMs are generative models that explicitly attempt to model a joint probability distribution $p(y, x)$ over observed features x (e.g., sensor inputs) and hidden state y (activities). An HMM requires two independence assumptions for tractable inference. The first assumption is that the future state depends only on the current state, not on past states – i.e., the hidden state at time t, y_t depends only on the previous hidden state y_{t-1}, or in other words,

$$P(y_t|y_1, ..., y_{t-1}) = P(y_t|y_{t-1})$$

The second assumption is conditional independence of observation parameters, i.e.,

$$P(x_t|y_t, x_1, ..., x_{t-1}, y_1, ..., y_{t-1}) = P(x_t|y_t).$$

To define the most probable hidden state sequence from an observed input sequence, the HMM finds a state sequence that maximizes the joint probability $p(x, y)$ of the transition probability $p(y_{t-1}|y_t)$ and the observation probability $p(x_t|y_t)$ – that is, the probability that x_t is observed in state y_t [9]:

$$p(x, y) = \prod_{t=1}^{T} p(y_t|y_{t-1})p(x_t|y_t)$$

Although there are advantages to the HMM approach, it also has its limitations; the dimensionality of x can be very large and the features can have complex dependencies, so constructing a probability distribution over them can be difficult [8]. Modelling the dependencies among input features can lead to intractable models, but ignoring them can lead to reduced performance.

B. Conditional Random Fields

CRFs are a solution to the mentioned problem in HMMs by modeling the conditional distribution $p(y|x)$ directly, which is all that is needed for classification. CRFs combine the ability to compactly model multivariate data with the ability to leverage a large number of input features for prediction. The advantage to a conditional model is that dependencies that involve only variables in x play no role in the conditional model, so that an accurate conditional model can have much simpler structure than a joint model [8]. Conditioning on the observations vastly expands the set of features that can be incorporated into the model without violating its assumptions.

Achieving high classification accuracy in complex tasks, such as activity recognition, often requires the use of domain knowledge to construct sophisticated features of the input observations. Such features typically incorporate information from more than a single time step. Features that span time steps violate the independence assumptions of the HMM, but not those of the CRF [6]. A CRF allows for arbitrary, dependent relationships among the observation sequences, and the hidden state probabilities can depend on past and even future observations. A CRF is modeled as an undirected graph, flexibly capturing any relation between an observation variable and a hidden state. They are thus especially suitable for classification tasks with complex and overlapped features. Figure 2 shows a CRF model.

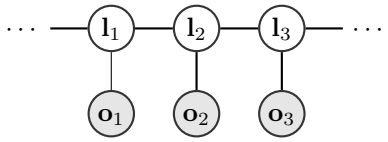

Fig. 2. An example of CRF where l_i represents the label of the hidden state and o_i is the observations such as sensor inputs.

After a CRF is specified, there are two essential tasks to do: training and inference. In training, the parameters of the model are learned from the labelled data instances. The goal is to determine the optimal weights of feature attributes such as observations and pairwise features.

The task of inference is to infer the hidden values (activity labels) from the observations (e.g., time stamp, location of people, etc.) Given a model of CRF including the set of features and their weights (the learned model), we first instantiate the CRF from the data instance, then we do inference over the instantiated CRF.

In the rest of this paper, we focus on CRFs for our activity recognition problem. We now illustrate a general approach called Virtual Evidence Boosting (VEB) proposed in [10] for training CRFs which addresses the issues of growing complexity in the CRF models. Later in this paper, we explain how we extend the VEB in modeling sequential multiple concurrent activities.

C. Training CRFs Using Virtual Evidence Boosting

Despite flexibility, training CRFs with large numbers of features is challenging [10]. Standard training algorithms based on Maximum Likelihood (ML) require running inference at each iteration of the optimization, which can be very expensive. Moreover, because the exact inference in general

Markov network including CRF is NP-hard, the approximate inference is often used.

An alternative method called Maximum Psuedo-Likelyhood (MPL) [11] suggests to convert a CRF into a set of independent patches; each patch consists of a hidden node and the true values of its direct neighbours (figure 3).

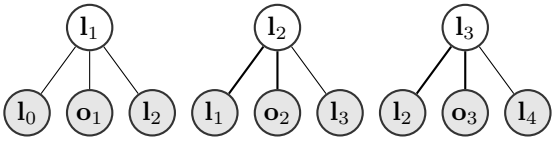

Fig. 3. Converted CRF for MPL learning by copying the true labels of neighbours as local evidence.

The ML estimation made on this simplified model works efficiently and has been successful in several domains, however, it has been observed to overestimate the dependency parameters in some experiments [12]. To overcome the limitation in MPL, a suggested solution is to treat the neighbour labels as virtual evidence or beliefs instead of observed. This is the main idea of Virtual Evidence Boosting (VEB) approach proposed in [10] which extends the standard boosting algorithm to handle input features that are either virtual evidences in the form of likelihood values or deterministic quantities. The extended boosting algorithm performs feature selection and parameter estimation in a unified manner and thus learns dependency structures in the relational data.

The algorithm extends Additive Logistic Regression (LogitBoost) [13] to handle virtual evidences or beliefs. That is, for each feature in the feature vector x_i, the input to boosting could be a probabilistic distribution over that feature's domain, as opposed to a single observed value.

D. Human Activity Recognition

Few studies have investigated the inference of concurrent and interleaved activities. Gu et al. [3] propose a data mining method using emerging patterns to extract unique patterns in activities of daily living in a home setting and tries to infer parallel activities of one person [3] or shared activities of multiple people [14] . An emerging pattern (EP) is a feature vector of each activity that describes significant changes between two classes of data. For instance, a feature vector {location is kitchen, object is burner} is an EP of a cooking activity and {object is cleanser, object is plate, location is kitchen} is an EP of a cleaning a dining table activity. A pattern is frequent if its support is no less than a predefined minimum support threshold. Coupled HMMs is another attempt by Wang et al. [2] to infer shared activities of multiple users in a home setting.

Dynamic CRFs and HMMs have been used for labelling multiple activities of an individual user. One study [15] has tried to recognize multiple activities by creating a fully connected Factorial CRF where each activity creates a chain and all hidden states share the same observations. This solution suffers from scalability problem. The complexity of this model increases dramatically as the number of activities and accordingly the number of chains increases. Another study [16] has built a skip chain CRF to recognize the interleaved activities combined with a correlation graph for recognition of concurrent activities. As the focus of the work is on a single user, it is assumed that a person's goals or activities are usually related, and based on how similar the activities are, they are more likely to appear together.

In all this research, parallel and interleaved activities of one or at most two people have been studied. In case of multiple people, only their shared activities such as watching TV together is recognized. The focus of these studies has mainly been activities of daily routines, often simplified to a few number of high level activities. Although some of them have mentioned targeting the problem of several people engaged in different activities as their future steps, to our knowledge no experimental results have been presented. The rest of this paper is focused on our proposed approach in sequential multiple concurrent activity recognition in shared settings.

E. Other methods for sequence classification

Several other methods have been proposed within the computational intelligence community for sequence classification, most prominently recurrent neural networks trained with backpropagation through time [17, 18], but also e.g. finite state machines evolved with genetic programming. A comparison with such non-structured techniques would be interesting future work, but is out of scope for the current paper.

III. Modeling Multiple Concurrent Activities

In the previous section, we formulated the activity recognition problem as a supervised learning approach. In this section, we present our model of activities that will be used in the classification.

Definition III.1. *A predefined set A of actions in form of $\{a_0, a_1, a_2, ..., a_k\}$ is defined as base-actions which are not mutually exclusive, as each instance can belong to more than one class.*

The classification [19] is a two step process: First, the classification algorithm builds the classifier from a training set. Then, the model is used to classify and label the unseen data into categories or classes in which instances most likely belong. In other words, a classification task is defined as follows: Let X be the set of instances to be classified, Y be the set of labels, and H be the set of classifiers for $X \rightarrow Y$. The goal is to find the classifier $h \in H$ maximizing the probability of $h(x) = y$, where $y \in Y$ is the ground truth label of x, i.e.,

$$y = argmax_i P(y_i|x)$$

Each instance is only assigned to one class and therefore classification errors occur when the classes overlap in the selected features, i.e., the same instance can belong to more than one class. This is the problem in case of concurrency where different actions are performed at the same time. For

example, in a surgical procedure if the action of 'intubation' (labelled e.g., a_7) occurs simultaneously with the action of 'surgical instrument preparation' (labelled e.g., a_5), then the corresponding data stream in the data set can be labelled as both a_7 and a_5. We address this issue by introducing *joint actions*.

Definition III.2. *Given the set* $A = \{a_0, a_1, a_2, ..., a_k\}$ *of base-actions, the set of joint-actions is of form* $\{ja_0, ja_1, ja_2, ..., ja_m\}$ *where* $ja_i \in \emptyset \bigcup A \bigcup A \times A \bigcup A \times A \times A \bigcup A \times A \times A \times A \bigcup ...$

In the mentioned example, a joint action (e.g., ja_1) is built by combining a_5 and a_7 if these two base-actions are observed together in a sequence.

Transforming multi-labelled data to single-labelled gives possibility to a more computationally efficient classification. In addition, more algorithms can be used for prediction. The important issue with joint classification is that the data belonging to joint-activity classes can be too sparse to build usable models. In our datasets, however, the joint-activities comprise over 70% of the dataset. We apply the Apriori algorithm [20] on our total data to find the pattern of joint actions. Apriori is a well-known mining algorithm to find frequent patterns in the data. Given a set of items, the algorithm attempts to find subsets with a minimum support. It uses a join and a prune step, where frequent subsets are extended one item at a time and any (k-1)-itemset that is infrequent cannot be a subset of a frequent k-itemset. The algorithm terminates when no further successful extensions are found.

IV. Modeling Temporal Dependency

Traditional classification methods assume independence between labels and can be considered insufficient in time series problems. Therefore, we need to extend the feature model to address temporal dependencies. We solve this issue by presenting additional types of information as evidence to the classifier.

Before going into details, we define the notion of observable feature in order to distinguish between this type and evidential types.

Definition IV.1. *A feature is observable if it can be sensed or obtained directly from the environment.*

Observable features include sensor inputs about e.g., tools and people.

A. Virtual Evidence

The value of a virtual evidence feature (ve) is computed using belief propagation (BP) which works by sending local messages through the graph structure. A message $m_{ij}(y_j)$ for each pair of neighbors y_i and y_j is a distribution sent from node i to its neighbor j about which state variable y_j should be in. The messages propagate through the CRF graph until they (possibly) converge, and the marginal distributions can be estimated from the stable messages. A complete BP algorithm defines how to initialize messages, how to update messages, how to schedule the order of updating messages, and when to stop passing messages. We explain the steps in sum-product algorithm [21]:

1) *Message initialization*: Usually all messages $m_{ij}(y_j)$ are initialized as uniform distributions over y_j.
2) *Message update rule*: The message $m_{ij}(y_j)$ sent from node i to its neighbor j is updated based on local potentials $\phi(y_i)$, the pairwise potential $\phi(y_i, y_j)$, and all the messages to i received from i's neighbors other than j (denoted as $n(i) \setminus j$). More specifically, for sum-product, we have
$$m_{ij}(y_j) = \sum_{y_i} \phi(y_i)\phi(y_i, y_j) \prod_{k \in n(i) \setminus j} m_{ki}(y_i)$$
3) *Message update order*: The algorithm iterates the message update rule until it (possibly) converges. Usually at each iteration, it updates each message once, and the specific order is not important (although it might affect the convergence speed).
4) *Convergence conditions*: To test whether the algorithm converges at an iteration, for each message, BP measures the difference between the old message and the updated one, and the convergence condition is met when all the differences are below a given threshold ε. More formally, the condition is
$$||m_{ij}(y_j)^k - m_{ij}(y_j)^{k-1}|| < \varepsilon, \forall i, \text{ and } \forall j \in n(i)$$
where $m_{ij}(y_j)^k$ and $m_{ij}(y_j)^{k-1}$ are the messages after and before iteration k, respectively.

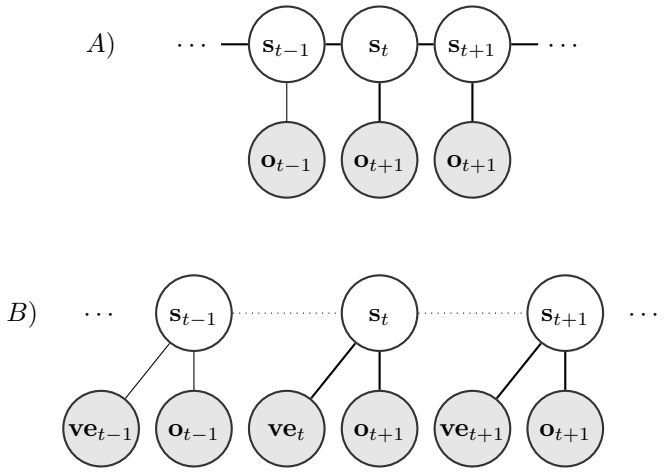

Fig. 4. Regular CRF structure (A) and wih added virtual evidence (B).

Figure 4 shows a regular CRF in (A) and the converted structure after adding virtual evidence in (B). The idea of VEB complements our model for adaptation in a computational level and we can indicate the past context that might matter for the situation in terms of past states distribution.

V. Activity Learning Models

This section presents our proposed learning models that incorporate evidential features to help address dependencies

between hidden states. For joint labelled actions, we first construct three different sub-models with added virtual evidence (ve). This approach extends VEB in two ways:

1) It is used for the learning of multiple joint actions.
2) It is used in dynamic CRF structures (multi-chained CRFs).

Activities can be modelled in different ways in the CRF framework. The VEB approach in [10] was tested on a single sequence of activities. In our approach for multiple concurrent activities, we extend the VEB to cover joint-actions and experiment with three models in order to find the best performing construction:

- Model A: Union joint labelling of multiple actions structured in a linear chain CRF with added virtual evidence from neighbour states as presented in figure 5.
- Model B: Team-based joint labeling where multiple actions of each team are modelled in parallel CRF chains with same set of observations for each state but separated virtual evidence. The model is shown in figure 6. Each chain is trained individually on the same data but they run in parallel to make inference on the test data. The results are then combined.
- Model C: Team-based joint labeling where multiple actions of teams are modelled in a coupled-chain CRF with same set of observations and virtual evidence for each state. This model addresses dependencies within and between teams (figure 7).

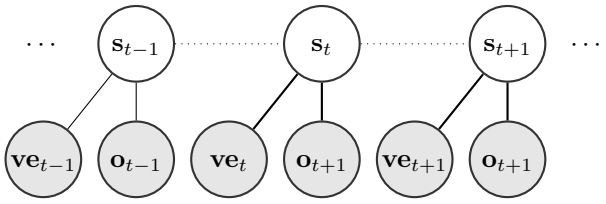

Fig. 5. Model A

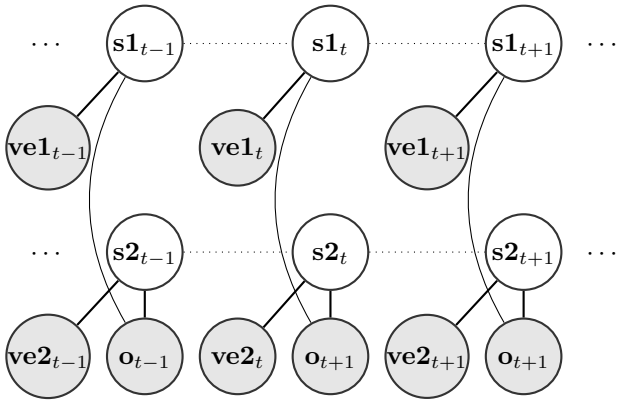

Fig. 6. Model B

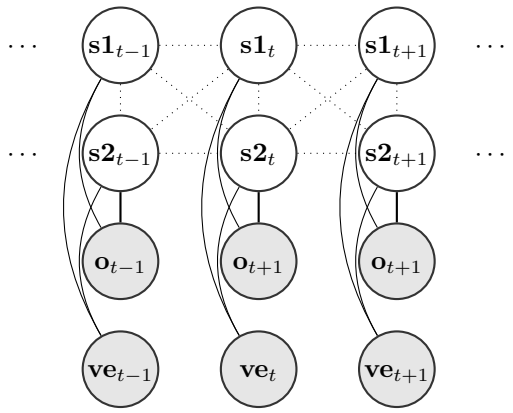

Fig. 7. Model C

VI. Collaborative activities in surgical rooms

Our general observations of surgical operations in different Danish hospitals reveal that in a typical surgery at least 6 clinicians with different specializations participate. The team includes at least one anaesthesia nurse mainly responsible for patient monitoring during the operation, an anaesthesiologist, a surgeon, a surgical assistant, a surgical nurse assisting with instruments, and a circulating nurse for general help and communication between inside and outside of the room. The surgical activity follows a temporal and sequential pattern:

The procedure usually starts with the anaesthesia team preparing devices, drugs and instruments followed by preparation of the patient for anesthaetization. While the patient is being anesthaetized, the surgical instruments and devices are prepared by surgical- and circulating nurses. After the patient is anesthaetized, he is prepared for the incision process. During the surgical execution, the patient's condition is monitored by the anesthaesia nurse. The procedure is finished after the cut is closed which means that the instruments and devices can be removed from the patient's body. The operation is considered as ended when the patient wakes up and is transferred to the recovery department.

Based on what distinguishes surgical actions from each other and allows them to be detected individually, we have identified a set of 17 main tasks which we refer to as base-actions. These base-actions are listed in Table I.

The surgical team members collaborate in parallel to carry out an overall task which entails performing a series of actions. Some of these actions can be done by only one clinician, such as monitoring the vital signs of the patient during surgery, while others involve several people, like the surgical procedure which at minimum involves the surgeon and the assisting surgical nurse.

Action	*Label*
a_1	Checking the anesthesia machine
a_2	Anesthetic preparation
a_3	Anesthesia-instrument preparation
a_4	Patient preparation for anesthesia
a_5	Surgical-instrument preparation
a_6	Anesthetization
a_7	Intubation
a_8	Preparing the patient for operation
a_9	Incision
a_{10}	Main procedure
a_{11}	Patient monitoring
a_{12}	Collecting surgical instruments
a_{13}	Waking the patient
a_{14}	Closing
a_{15}	Cleaning up
a_{16}	Extubation
a_{17}	Recovery ready

TABLE I

THE LIST OF BASE-ACTION TYPES IN AN OR

	PM_1 (accuracy)			PM_2 (time taken)		
Dataset	*A*	*B*	*C*	*A*	*B*	*C*
1	19,0	45,7	46,8	370	44,5	108
2	44,2	69,2	62,8	591	55,5	480
3	36,7	61,2	54,2	324	56	184
4	43,2	65,9	66,5	404	56,5	495
5	40,5	59,5	49,4	288	76,5	468
6	66,0	77,8	78,5	316	34	130
7	62,0	75,9	80,4	361	39,5	75
8	44,4	50	58,7	362	29,5	197
9	48,0	68,5	68,5	416	36	960
10	31,2	57,7	54,5	352	27	220
	43,1	**63,1**	61,9	378,4	**45,5**	331,7

TABLE II

PERFORMANCE RESULTS OF TRAINING AND TEST IN DIFFERENT MODELS WITH PAIRWISE RELATIONS BETWEEN NEIGHBORS. FOR EACH PERFORMANCE MEASURE (1 AND 2) ALL THREE MODELS (A, B AND C) ARE TESTED. THE HIGHEST ACCURACY AND SHORTEST INFERENCE TIME IS OBTAINED IN MODEL B.

A. Data Collection Using a Sensor Platform

Based on our detailed study, the following parameters were important to track:

- The location of clinicians
- The location of objects on different tables
- The use of objects and instruments by the clinicians

We created a sensor platform with three sub-sensor systems sensing each of the items listed above, and a central server for collecting, filtering, time stamping, synchronizing, and storing sensor readings [22]. The setup included palm-based sensors, table-based sensors, and the Ubisense location tracking system. We tagged real surgical instruments and performed the operations on a fictive patient.

The scenarios were based on the video recorded operations and designed in close collaboration with domain experts, i.e. surgeons, anaesthesiologists, and nurses. We annotated data from 10 simulated laparoscopic operations. The total length of the dataset was 11823 instances which were created once a second.

B. Feature extraction and labeling

A preliminary step was to convert sensor inputs to binary values and subsequently make a feature instance based on all relations acquired at a time stamp. The raw sensor readings were sampled, synchronized and transformed into feature instances by the sensor platform on the fly. We used an annotation and verification tool to annotate the data and also to check the sensor readings. By applying the Apriori on our data, we identified the pattens of joint-actions. The joint cardinality which is the length of possible joint-activities observed in the data ranged from 1 to 5. The results showed that more than 70% of the instances have a joint-label of length ¿2, i.e., at least 70% of the time, more than one activity occurs in the OR and at each time stamp t up to 5 concurrent activities can be observed.

VII. THE LEARNING PROBLEM

We were interested in evaluating the performance of the proposed techniques both in terms of the accuracy of the inferred labels and the inference time. The latter is important as the activity recognition is expected to be done in real-time. Hence, the learning problem is defined as:

Task T: Recognition of concurrent team actions in an OR
Performance measure PM_1: Correctly classified actions
Performance measure PM_2*:* The inference time
Training Experience E: Using data from surgical operations

As is common in machine learning, we divided datasets into a training part and a testing part. The training dataset is used for building the classifier, and the testing dataset is used for accuracy evaluation. We used the leave-one-out cross validation method on 10 datasets (10 operations), where we each time trained the classifier on 9 operations and tested on the last one. The final result is the average from 10 experiments. The main metric in our evaluation is accuracy which is defined as the number of testing instances accurately classified divided by the number of testing instances.

The primary CRF structure covered pairwise relations between two neighbors of each node, i.e., the previous and the immediate next node. We experimented with three construction models described in the last section [1]. Table II, figures 8, and 9 summarize performance results of different models. In the following, we describe the details of applying each model on the data.

A. Model A – Union Joint Labeling and Union Training

In model A (figure 5), the total number of labels was 65 which is quite high compared to the size of datasets, so this setting did not scale. As shown in table 8, this model performs slowly and the accuracy is lowest.

[1]The CRF was implemented in C++

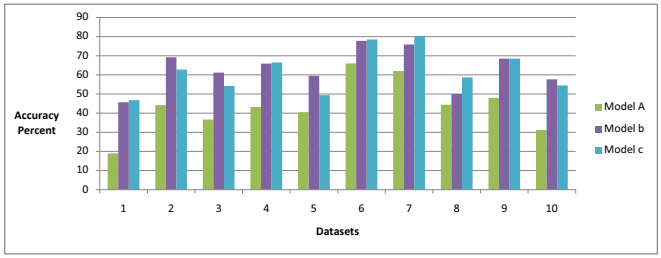

Fig. 8. Comparing accuracy measure for models A, B, and C during cross validation on 10 datasets.

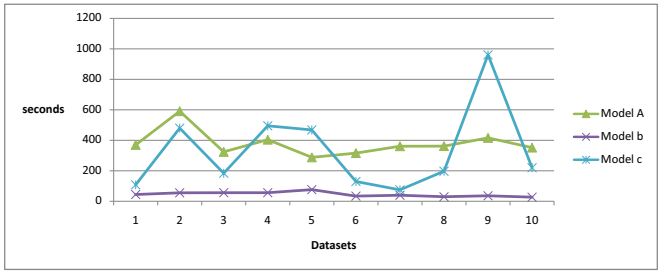

Fig. 9. Comparing differences in the inference time in models A, B, and C during cross validation on 10 datasets.

B. Model B – Team Based Labeling and Individual Training

The scalability problem in model A led to building model B, where we divided the actions of the anesthesia and operating team and did the joint labelling on each team's actions. We then instantiated a CRF for each team based on the data. In model A, the number of joint labels was large and it caused the model to perform slowly and less accurate. Individual labelling of team actions decreased the number of labels and hence improved the inference time as well as the accuracy (see table II). On average, the accuracy increased by 20% to 63% compared to model A and the inference time fell from 378 seconds to 45 seconds.

C. Model C – Team Based Labeling and Coupled Training

As mentioned earlier, there is temporal dependencies between OR actions both within and across teams. Separate training of two team actions only models the dependencies within each team. In order to capture pairwise relations across teams, we built a coupled (two-chains) CRF model of team actions (see figure 7), where the set of observable features was shared between the chains and pairwise relations between nodes were specified. Three types of pairwise relations were defined:

- Anesthesia - Anesthesia
- Surgical - Surgical
- Anesthesia - Surgical

As shown in table II, the average accuracy is about 62 percent which is almost as high as with model B, but the average inference time ($\simeq$ 332 seconds) is almost seven times longer which is caused by a double number of nodes and more complex pairwise relations. However, this model outperforms the model A in both measures.

D. A note on accuracy

While an accuracy of 63%, it should be noted that the baseline obtained from majority guess (a dumb classifier that always outputs the most common joint activity) is 22%; Given the very high number of classes, the random guess baseline is $100/65 = 1.53\%$. As the intended application for the activity recognition performed here is recommender systems, very high accuracy is not necessary, though obviously desirable.

VIII. Discussion and Conclusion

This paper presented our approach to sequential and multiple concurrent activity recognition. We discussed existing machine learning methods, especially the two most widely used methods in sequence classification for ubiquitous computing, HMM and CRF. We chose the CRF framework to experiment with due to its flexibility in addressing pairwise relations between states and features. We extended the VEB approach with three models to be used for recognition of multiple concurrent activities. We found that model B achieved both the best classification accuracy and the shortest inference time. The accuracy is acceptable for issuing meaningful recommendations. However, the performance of the models suffers from the iterative boosting steps where the virtual evidence is computed by the BP method. Especially in cases of a long sequence and a large number of hidden states the computational time for sending messages increases which results in a long learning and inference time. As a remedy, we propose historical evidence which is extracted from observable features as well as neighbor labels using aggregate functions. The aggregated values are stored as evidence features and then included in the classification.

Future research includes using historical evidence markers as a complement or alternative to virtual evidence, and using the activity recognition techniques and models presented here to recommend relevant information and activities for clinical teams in real-time; initial work on this is presented in [23]. To the extent possible, the techniques should also be tested on more diverse clinical data obtained from embedded sensors during real surgical operations. Another future study will be to apply our method using HMMs and compare the results with CRFs.

The data used in this experiment was collected in a simulated setup which raises the question of whether or not this data represents the real world scenarios in an operating room. Although most operations share common types of activities and tasks, such as anesthaetization and incision, we agree that some unique activities might be required in some operations and unexpected critical situations might arise. However, the focus of this study has been on recognition of general concurrent activities in the OR, and in that sense, we think the data used in this experiment is generic enough to address the problem.

For more technical details on the current study, including parameter studies, please see the first author's PhD thesis [24].

References

[1] J. Bardram and A. Friday, *Ubiquitous Computing Systems*. Taylor & Francis Group, 2010, ch. 2, pp. 37–94.

[2] L. Wang, T. Gu, X. Tao, and J. Lu, in *Ambient Intelligence*, Berlin, Heidelberg, ch. 10.

[3] T. Gu, Z. Wu, X. Tao, H. K. Pung, and J. Lu, "epsicar: An emerging patterns based approach to sequential, interleaved and concurrent activity recognition," in *PerCom*, vol. 0. Los Alamitos, CA, USA: IEEE Computer Society, 2009, pp. 1–9.

[4] Y. Wang, X. Hou, and T. Tan, in *Computer Vision – ACCV 2006*, P. J. Narayanan, S. K. Nayar, and H.-Y. Shum, Eds. Berlin/Heidelberg: Springer-Verlag Series = Lecture Notes in Computer Science, Title = Recognize Multi-people Interaction Activity by PCA-HMMs, Volume = 3851, Year = 2006, Bdsk-Url-1 = http://dx.doi.org/10.1007/11612032

[5] D. Choujaa and N. Dulay, in *2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*.

[6] D. L. Vail, M. M. Veloso, and J. D. Lafferty, "Conditional random fields for activity recognition," in *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, ser. AAMAS '07. New York, NY, USA: ACM, 2007, pp. 235:1–235:8. [Online]. Available: http://doi.acm.org/10.1145/1329125.1329409

[7] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," in *Proceedings of the IEEE*, no. 77, 1989, pp. 257–286.

[8] C. Sutton and A. McCallum, "An Introduction to Conditional Random Fields for Relational Learning," in *Introduction to Statistical Relational Learning*, L. Getoor and B. Taskar, Eds. MIT Press, 2007. [Online]. Available: http://www.cs.berkeley.edu/ casutton/publications/crf-tutorial.pdf

[9] E. Kim, S. Helal, and D. Cook, "Human activity recognition and pattern discovery," *IEEE Pervasive Computing*, vol. 9, pp. 48–53, 2010.

[10] L. Liao, T. Choudhury, D. Fox, and H. Kautz, "Training conditional random fields using virtual evidence boosting," in *IJCAI'07: Proceedings of the 20th international joint conference on Artifical intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007, pp. 2530–2535.

[11] J. Besag, "Statistical Analysis of Non-Lattice Data," *Journal of the Royal Statistical Society. Series D (The Statistician)*, vol. 24, no. 3, pp. 179–195, 1975. [Online]. Available: http://dx.doi.org/10.2307/2987782

[12] C. J. Geyer and E. A. Thompson, "Constrained Monte Carlo Maximum Likelihood for Dependent Data." [Online]. Available: http://www.jstor.org/stable/2345852

[13] J. Friedman, T. Hastie, and R. Tibshirani, "Additive Logistic Regression: a Statistical View of Boosting," *The Annals of Statistics*, vol. 38, no. 2, 2000.

[14] T. Gu, Z. Wu, L. Wang, X. Tao, and J. Lu, "Mining emerging patterns for recognizing activities of multiple users in pervasive computing," in *Mobile and Ubiquitous Systems: Networking Services, MobiQuitous, 2009. MobiQuitous '09. 6th Annual International*, july 2009, pp. 1 –10.

[15] T.-Y. Wu, C.-C. Lian, and J. Y. Hsu, "Joint Recognition of Multiple Concurrent Activities using Factorial Conditional Random Fields," in *2007 AAAI Workshop on Plan, Activity, and Intent Recognition, Technical Report WS-07-09. The AAAI Press, Menlo Park*, 2007.

[16] D. H. Hu and Q. Yang, "Cigar: concurrent and interleaving goal and activity recognition," in *AAAI'08: Proceedings of the 23rd national conference on Artificial intelligence*. AAAI Press, 2008, pp. 1363–1368.

[17] J. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[18] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

[19] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques, Second Edition (The Morgan Kaufmann Series in Data Management Systems)*, 2nd ed. Morgan Kaufmann, Jan. 2006.

[20] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proceedings of the 20th International Conference on Very Large Data Bases*, ser. VLDB '94.

[21] L. Liao, "Location-based Activity Recognition," Ph.D. dissertation, University of Washington, 2006.

[22] J. Bardram, A. Doryab, R. Jensen, P. Lange, K. Nielsen, and S. Petersen, "Phase recognition during surgical procedures using embedded and body-worn sensors," in *2011 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, march 2011, pp. 45 –53.

[23] A. Doryab, J. Togelius, and J. Bardram, "Activity-aware recommendation for collaborative work in operating rooms," in *Proceedings of the ACM conference on Intelligent User Interfaces (In press)*, ser. IUI '12. ACM, 2012.

[24] Afsaneh Doryab, "Context-aware Information Adaptation In Collaborative Settings," Ph.D. dissertation, IT University of Copenhagen, Denmark, October 2011.