Searching for Temporal Patterns in AmI Sensor Data

Romain Tavenard[1], Albert A. Salah[2], and Eric J. Pauwels[2]

[1] IRISA/ENS de Cachan, Av. du Gén. Leclerc, 35042 Rennes Cedex
[2] CWI, Kruislaan 413, 1098 SJ Amsterdam, NL

Abstract. Anticipation is a key property of human-human communication, and it is highly desirable for ambient environments to have the means of anticipating events to create a feeling of responsiveness and intelligence in the user. In a home or work environment, a great number of low-cost sensors can be deployed to detect simple events: the passing of a person, the usage of an object, the opening of a door. The methods that try to discover re-usable and interpretable patterns in temporal event data have several shortcomings. Using a testbed that we have developed for this purpose, we first contrast current approaches to the problem. We then extend the best of these approaches, the T-Pattern algorithm, with Gaussian Mixture Models, to obtain a fast and robust algorithm to find patterns in temporal data. Our algorithm can be used to anticipate future events, as well as to detect unexpected events as they occur.

1 Introduction

The success of Ambient Intelligence (AmI) depends on observing the activities of humans and responding to their behaviour patterns intelligently. In ubiquitous environments, where a wealth of sensory data is produced, mining the data for temporal patterns serves this need by discovering associations and structure, either in an offline manner to pave the way for new designs and applications, or in an online manner to ensure adaptation to the user of the AmI environment.

Two things make this task especially challenging. First of all, in a real environment, action patterns that are composed of separate events are interleaved, either by the presence of multiple users, or simply by our habit of doing multiple actions at the same time. Thus, taking an event window to predict the next event in the system will simply not work. Secondly, these patterns exist in different time scales, and the time difference between related events of a single action can have a large variation. Consequently, detecting associations with these patterns becomes a very challenging task, and many traditional pattern analysis methods are not directly applicable as we show in the next section.

In Section 2, a brief survey of the relevant literature is presented, with an emphasis on the more prominent compression-based approaches. The T-pattern method and our proposed modifications to it are presented in Section 3 and Section 4, respectively, followed by our experimental results.

2 Description of the Problem and Related Work

The temporal data we would like to analyze is in form of a sequence of point events, derived from a dense network of non-intrusive and low-resolution sensors. The patterns that we hope to detect are in the form of short event sequences, with additional information about the expected time of each event in the sequence, relative to the previous event. These patterns can then serve for semantic analysis or prediction of events for responsive environments, for instance in scheduling of maintenance jobs or in arming home security systems. The challenge of the problem is the existence of multiple causes (e.g. multiple users of the environment), triggering unrelated events one after the other.

The most straightforward way to detect temporal events is by representing them spatially, where portions of the input feature are associated with increasing time indices. This approach does not work except for the simplest cases, as the absolute positions in a feature vector are not relevant at all.

A more appropriate way of representing time is to make it a part of the model. For instance in recurrent neural networks, the temporal dimension is taken into account with the help of context units [3]. However, recurrent neural networks and related approaches cannot deal with overlapping patterns, they quickly become cumbersome for larger input intervals, and they require lots of training samples.

Markov models have been recently employed to tackle simplified versions of this problem, where there are no action overlaps, and events are generated as one long sequence [6]. These models have three main disadvantages for the problem at hand. First and foremost, the first order Markovian assumption does not hold, as action patterns are construed as sequences of events, and the complete sequence is relevant for the prediction of the next event. Secondly, the estimation algorithms assume that the topology of the HMM-structure is known, which is not the case. Finally, they cannot predict patterns that have long event intervals.

A recent approach involves PCA-based methods to uncover daily human behaviour routines [2]. The data for each subject are stored in an activity matrix, whose most prominent eigenvectors (dubbed *eigenbehaviors*) are then interpreted. One obvious drawback with this method is that it requires a fixed sized activity vector. Additionally, there is no hierarchical decomposition of activities.

Finding a *dictionary* of patterns is possible with **compression-based algorithms** that treat events as "words" in a stream, and seek the patterns that lead to the best compression of the stream. These methods use the Lempel-Ziv compression algorithm, which is known to achieve Markov entropic compression, or a variant of it (e.g. Lempel-Ziv-Welch and Active Lempel-Ziv algorithms) [1].

The basic Lempel-Ziv algorithm (**LZ78**) uses an automatically updated dictionary to extract recurring "words" (patterns) in a string. The Lempel-Ziv-Welch (**LZW**) variant starts off with a pre-defined basic dictionary (in the case of sensor networks these are single sensor events) to avoid ill-detected patterns at the beginning of the stream and to introduce some continuity. The **Active LeZi** uses a sliding window of length l (length of the longest phrase in LZ table) on the stream to extract all possible sequences of size l.

LZW and Active LeZi both aim at adding continuity to LZ pattern extraction, yet they still have linear complexity, which is a beneficial feature for a real-time event detection system. On the other hand, none of the compression based methods take into account the temporal structure of the patterns, as the time delays are not modeled, and subsequently overlapping events may escape detection. For a dense, low-cost sensor network without the identification of event source, this is a major drawback as is clearly borne out by the experimental results reported below. This is the main reason why we turn our attention to *T-patterns* as discussed in the next section.

3 T-patterns

The temporal pattern detection methods mentioned in the related work section ignore the time information, and cast the problem into a simpler representation by retaining only the order of events. In neural network, HMM, and compression based approaches, the emphasis is on predicting the next event, which is not a suitable perspective for an environment with multiple overlapping event sequences.

In the *T-pattern* approach, as introduced and explored by Magnusson, symbolic time series are investigated, where each symbol represents the onset of a particular event or activity, with the principal goal of elucidating possible relationships between pairs of symbols and then building trees of temporal dependencies in a hierarchical fashion [5]. A thorough search is conducted on the training sequence for symbols of an ever-growing dictionary. As the algorithm proceeds, pairs of strongly correlated events joined into new events, and the search is resumed with the expanded dictionary.

Magnusson introduced the notion of a *critical interval* (CI): $[d_1, d_2]$ is considered to be a CI for the pair of symbols (events) (A, B) if the occurrence of A at time t entails that B is more likely to occur in the time interval $[t+d_1, t+d_2]$ than in a random interval of the same size. He then suggested to use the standard p-value to gauge how exceptional the observed frequency of the combination under scrutiny is.

More precisely, suppose the total data stream has length T with N_A and N_B occurrences of A and B, respectively. If we assume (following Magnusson [5]) as null-hypothesis that A and B are independent Poisson processes with intensity (i.e. the average number of events per unit time interval) $\lambda_A = N_A/T$, and $\lambda_B = N_B/T$, respectively. Now, assume in addition that there are N_{AB} occurrences of B in a predefined CI (of length d) trailing each A-event. Notice that under the null-hypothesis the expected number of B-events in a time interval of length d equals $\mu_B = \lambda_B d$. In particular, the probability of not observing a B-event in this CI is therefore equal to $\pi_0 = e^{-\mu_B} = e^{-\lambda_B d}$. The above-mentioned p-value is then computed as the probability of observing at least N_{AB} B-events in the CI, if we assume that A and B are independent. Hence,

$$p = P(N_{AB} \text{ B-events or more} \mid \text{A, B indep.})$$

$$= 1 - \sum_{k=0}^{N_{AB}-1} \binom{N_A}{k} (1-\pi_0)^k \pi_0^{(N_A-k)}.$$

Magnusson suggests as a T-pattern detection scheme to test for every possible pair of symbols of the form (A, B), every possible CI, from the largest to the smallest one, until the p-value is sufficiently small, indicating significance (.05 is a typical upper bound). Note that p will be high for high values of d, which means that short intervals will be favored.

4 The Modified T-Pattern Algorithm

We propose two modifications to the T-pattern algorithm to make it more resilient to spurious patterns, and to make the search for patterns more robust.

4.1 Testing independence between two temporal point processes

The repeated significance testing of the basic T-pattern approach substantially increases the risk of false positives (suggesting spurious dependencies). Applying a Bonferroni correction would be one way to mitigate this adverse effect. In this paper, however, we put forward a more efficient way of testing this independence between A and B which is based on the following proposition.

Proposition 1 *If A and B are independent temporal point process, then*

$$T_{AB} \sim U(0, \tilde{T}_B).$$

In plain language this proposition asserts that if the A and B processes are independent, then whenever an A-event occurs between two successive B-events, it will be uniformly distributed in that interval. Due to lack of space we will not attempt to give a rigorous proof, but it is intuitively clear that non-uniformity of A within the B-interval, would allow a keen observer to improve his or her prediction of the next B-event, thus contradicting independence (for a graphical illustration of this proposition we refer to Fig.2. This therefore allows us to formulate a statistical procedure to test whether A and B are dependent: using the notation established above we compare for each event A_k the time till the next B-event to the current B-interval length:

$$U(k) = \frac{T_{AB}(k)}{\tilde{T}_B(k)} = \frac{B_{k^*} - A_k}{B_{k^*} - B_{k^*-1}}$$

which, under the assumption of independence, should be uniformly distributed between 0 and 1: $U \sim U(0, 1)$. This can be easily checked by any number of standard statistical test (e.g. Kolmogorov-Smirnov). If the null hypothesis (independence) is rejected then it makes sense to start looking for inter-event time intervals (i.e. CI's). This is taken up in the next section.

4.2 Modelling T_{AB} times

The CI detection scheme as proposed in [5] has the drawback that only the first occurrence of B following A is considered. However, if the average occurrence rate of A is relatively high, or if the inter-event time for B is long, this could lead to fallacious associations.

For this reason, we propose to proceed differently. If the above-discussed uniformity test has rejected independence, then we look for the characteristic period by modelling the conditional probability $P(B \text{ at } t + \Delta t \,|\, A \text{ at } t)$ using Gaussian Mixture Models (GMM). More precisely, all the A-events are aligned at time zero, whereupon all subsequent B-events are plotted. If an A-event tends to induce a B-event after a delay of t time-units, this will show up in this plot as a significant peak. All the non-related B-events will contribute to a very diffuse background. For that reason, we model the B-events as a 2-component GMM. One sharp and localized peak sits on top of the critical interval, while all the other B-events give rise to a flat and broad second component. The standard variation of the sharp peak immediately suggest a value for the width of the CI.

5 Experimental Results

In order to have a simple and realistic experimental setup, we simulate simple interruption sensors in a home or office environment (See Fig.1). We have one or two users of the system generating simultaneous interruption events from a predefined event dictionary, which serves as a catalogue of prominent behaviours.

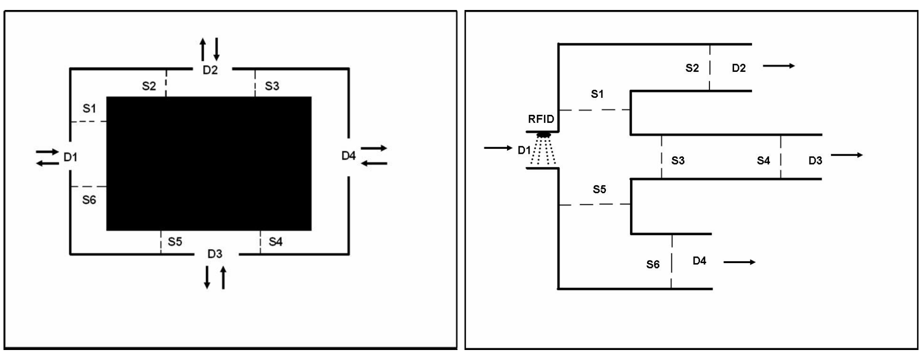


Fig. 1. Ground plan for two corridor layouts used in experiments. Left: Layout 1 shows 4 doors and 6 interruption sensors. Right: Layout 2 shows an entrance door and 3 exit doors, as well as an RFID reader and 6 binary interruption sensors.

For each configuration, we generated training and test sequences of $1,000$ symbols by simulating one or two persons walking in the corridors. We have investigated to what degree we could use the patterns discovered in the training phase as predictors for events in the second stream. The prediction is made for each discrete time slot, which is more granular than predicting the next event. We have contrasted the compression based methods, T-patterns, and our modified T-pattern approach. As the first symbol emitted by each new pattern is random and therefore completely unpredictable, and as individual patterns are short, the prediction rate will have an inherent upperbound.

Fig.3 summarizes the experimental results for the different compression algorithms. In each case the x-axis represents the minimal confidence in the prediction. Confidence is high (in fact 100%) whenever the current pattern unambiguously predicts a unique symbol. If there more potential outcomes, confidence drops accordingly. The dotted line indicates the percentage of cases for which prediction is possible with the confidence specified on the x-axis. The left column show results for scenarios in which only one person is present, the right column shows results for the case when two person intermingle. The two horizontal lines indicate the upper bound for the achievable accuracy (recall that the first symbol in each pattern is unpredictable) and the accuracy of a random prediction.

The results displayed in Fig.4 show contrasts the original CI-extraction (as detailed in [5]) with T-Patterns that use the GMM modelling expounded above. It transpires that Magnusson's original scheme produces too many (spurious) T-patterns making high-confidence prediction impossible as is clear from the way the curves quickly drop to zero. This is most apparent in the 2-person scenario where the intermingling of 1-person patterns generates a large number of new combinations, a fair bit of which are erroneously identified as T-patterns. The GMM approach fares much better, even in the more difficult 2-person scenario.

6 Conclusion

Detecting temporal patterns in sensor data is useful for semantic analysis and event prediction in AmI environments. In this paper we have reviewed two methodologies for the discovery of temporal patterns. The first one collapses the sequence into a string and then uses compression-based techniques to extract repetitive "words". The second one (so-called T-patterns) takes advantage of the time dimension to find the typical delay between related events. We have proposed some improvements to the basic T-pattern methodology (referred to in this text as GMM T-patterns) that significantly improve the performance. Experiments show that T-patterns outperform the compression-based techniques, which is not really surprising as the compression discards most of the temporal information. The experiments also show that the proposed T-pattern improvements (independence testing and GMM-modelling of correlation times) yield more reliable results.

To conclude we summarize the experimental results in Table 1. It was obtained by computing for each experiment the correct prediction rate for a confidence level of 20% (this amounts to constructing a vertical line at the x-value 0.20 in each of the figures and reading of the intersection with the solid curve). The significance of the proposed improvements is obvious.

	Layout 1		Layout 2	
	1 person	2 persons	1 person	2 persons
LZ	29.8	17.7	56.5	13.2
ALZ	21.1	18.8	66.4	19.6
LZW	28.9	22.0	60.5	15.1
T-patterns	28.8	17.1	61.5	24.2
GMM T-patterns	**34.8**	**29.3**	**61.9**	**48.3**

Table 1. Percentage correct predictions at the 20% confidence level.

While we focus on detecting behaviour patterns, a complementary problem would be to track multiple people using low-cost sensors, for which Bayesian filtering techniques are proposed in the literature [4]. The patterns that we aim to detect can serve the tracking problem in constructing a Voronoi graph of the environment. This application is currently inspected by our group.

Acknowledgments

This work was done while R.T. was at CWI. The authors gratefully acknowledge partial support by EC's NOE MUSCLE (FP6-507752), and the Dutch BSIK/BRICKS project. We are grateful to M.S. Magnusson for providing us with additional information on T-patterns.

References

1. Diane J. Cook. Prediction algorithms for smart environments. In Diane J. Cook and Sajal K. Das, editors, *Smart Environments: Technologies, Protocols, and Applications*, page 175. Wiley Series on Parallel and Distributed Computing, 2005.
2. N. Eagle and A. Pentland. Eigenbehaviors: Identifying structure in routine. In *Proc. Roy. Soc. A (in submission)*, 2006.
3. J.L. Elman. Finding Structure in Time. *Cognitive Science*, 14(2):179–211, 1990.
4. D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello. Bayesian Filtering for Location Estimation. *IEEE Pervasive Computing*, pages 24–33, 2003.
5. M. S. Magnusson. Discovering hidden time patterns in behavior: T-patterns and their detection. *Behav. Res. Methods Instrum. Comput.*, 32(1):93–110, February 2000.
6. S.P. Rao and D.J. Cook. Predicting inhabitant action using action and task models with application to smart homes. *International Journal on Artificial Intelligence Tools*, 13(1):81–99, 2004.

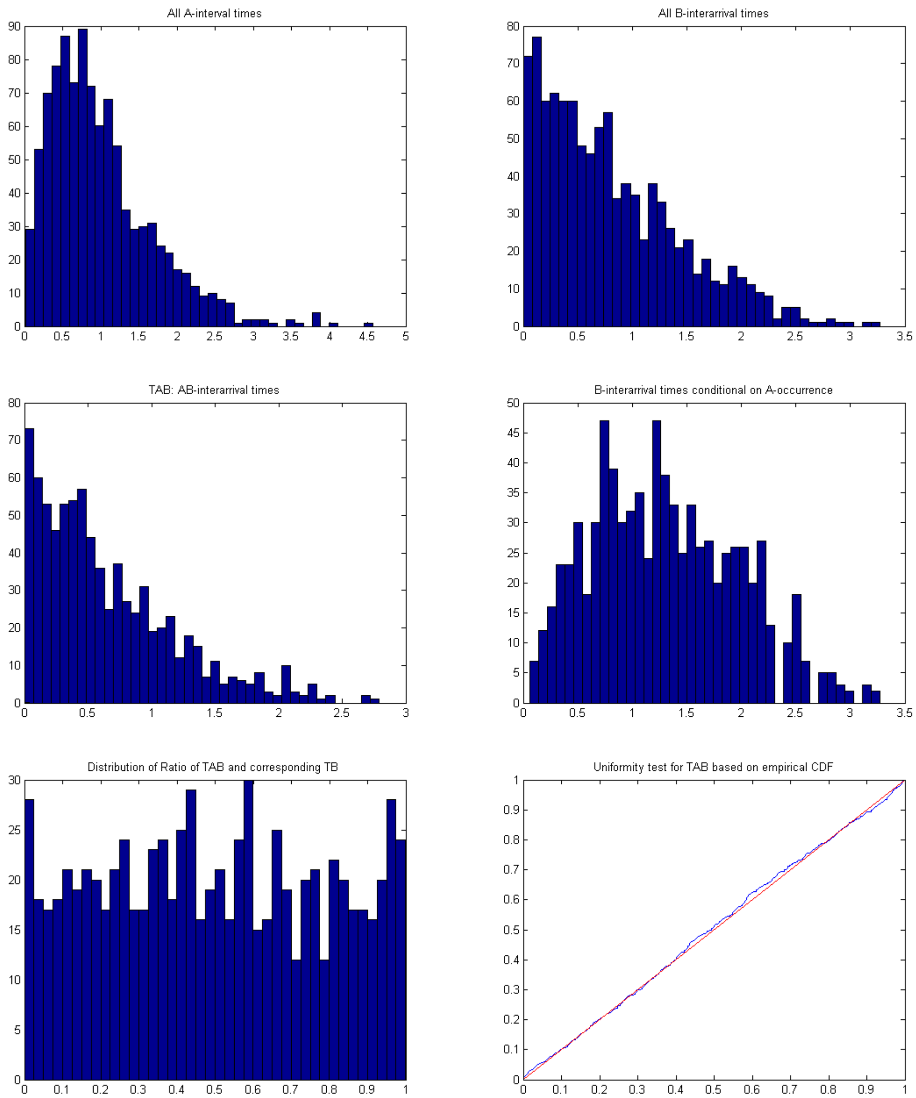
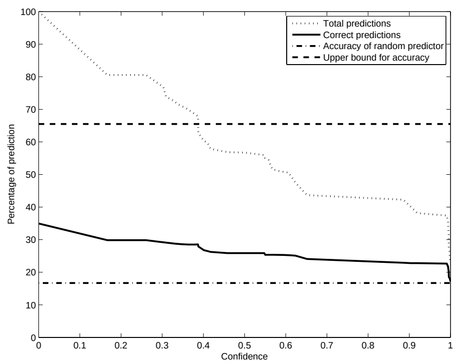
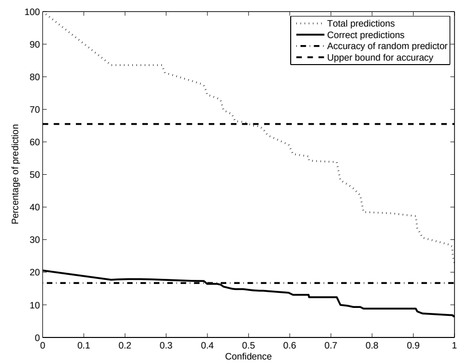


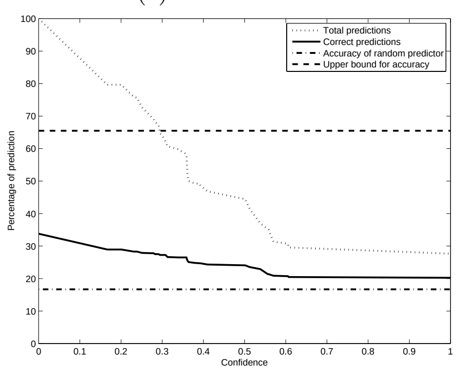
Fig. 2. *Top row:* Histogram for interevent times for the A (*left*) and B (*right*) process; *Middle row: left* T_{AB} distribution: time intervals between the occurrence of A and *next* B-event; *right:* Lengths of B-intervals in which a A-event occurred; notice the bias towards longer intervals (compared to histogram of all B interevent times above). *Bottom row: left:* Histogram of ratio $T_{AB}/\tilde{T}_B$, if A and B are independent, this ratio should be uniformly distributed between 0 and 1, a fact which is even more clearly borne out by its cumulative density function to the theoretically predicted one. (The p-value in this case was 0.61 which means that the null-hypothesis of independence is accepted.)



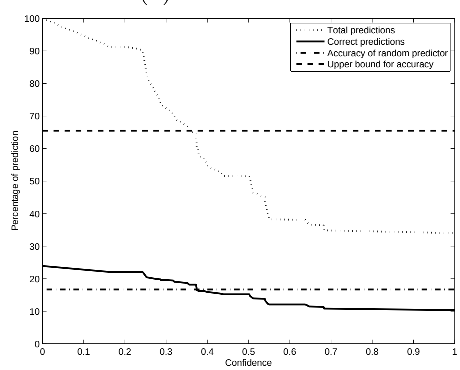
(a) LZ - 1 Person



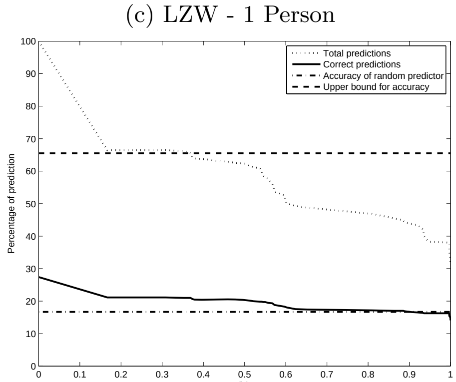
(b) LZ- 2 Persons



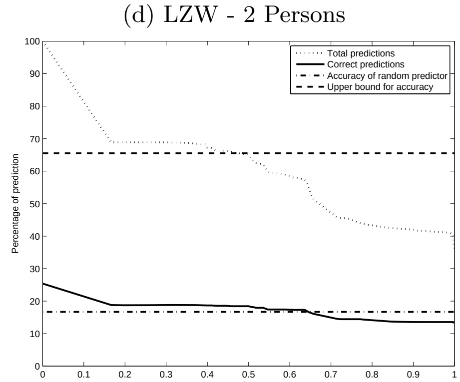
(c) LZW - 1 Person



(d) LZW - 2 Persons

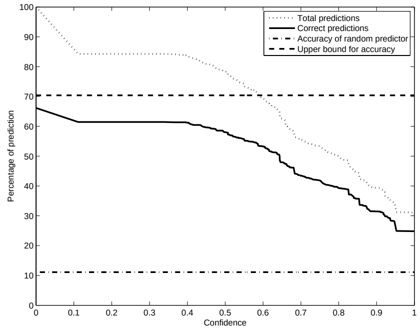


(c) ALZ - 1 Person

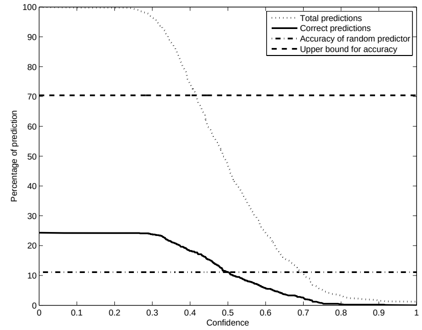


(d) ALZ - 2 Persons

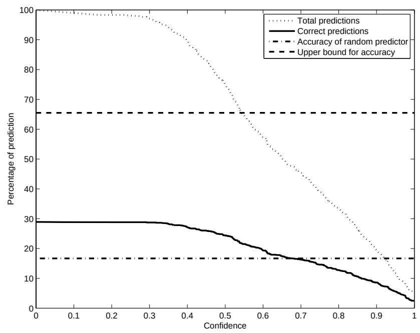
Fig. 3. Predition results for the compression based algorithms for the first layout. Detailed results for the second layout omitted for brevity.



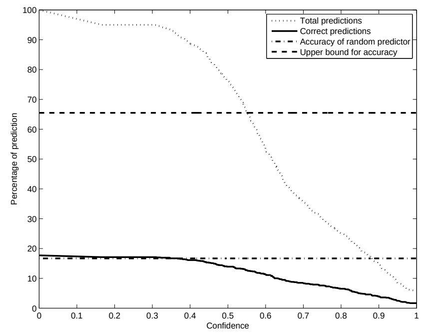
(a) Layout 1 - 1 Person



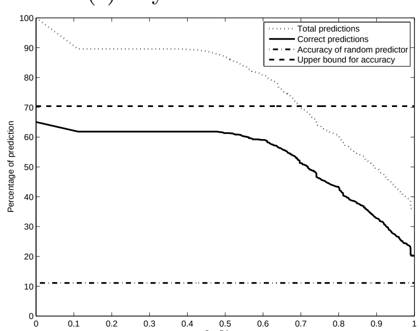
(b) Layout 1 - 2 Persons



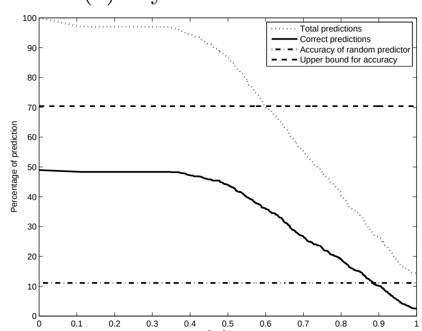
(c) Layout 2 - 1 Person



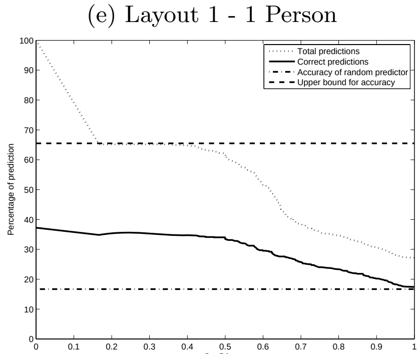
(d) Layout 2 - 2 Persons



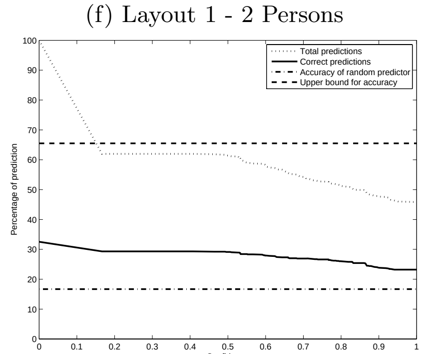
(e) Layout 1 - 1 Person



(f) Layout 1 - 2 Persons



(g) Layout 2 - 1 Person



(h) Layout 2 - 2 Persons

Fig. 4. Predition results for (a)-(d) the T-Pattern and (e)-(h) the modified algorithm