OPPORTUNISTIC SENSING AND LEARNING IN SENSOR NETWORKS

R. Tavenard[*]

IRISA, ENS de Cachan
Avenue du Général Leclerc
35042 Rennes Cedex, France

O. Ambekar, E.J. Pauwels, M. Waaijers

CWI, PNA4
Kruislaan 413
1098 SJ Amsterdam, The Netherlands

ABSTRACT

In this paper we will argue that sensor networks in which multimodal sensors are connected to each other and computational devices capable of data mining, offer the possibility of serendipitous and opportunistic sensing in which unanticipated associations are detected and used to produce more robust event recognition. To illustrate this point of view, we outline two simple scenarios and conduct some preliminary experiments to show that this type of on-the-fly data analysis can indeed be useful.

1. INTRODUCTION AND RELATED WORK

1.1. Opportunistic Sensing in Sensor Networks

The challenge of creating smart environments for ambient intelligence using pervasive or ubiquitous computing has spurned a lot of research on sensor networks and the analysis of their output. Most work focusses on sensor networks that have been set up with a specific task in mind and are therefore relatively focussed ("narrow-minded" if you like) in the data they monitor and the corresponding events they detect and report upon. However, with sensor availability and connectivity increasing by leaps and bounds, it has become viable to take a more experimental and "open-minded" stance and allow a network of sensors and computational devices to pro-actively inspect the many data streams that impinge on it in an effort to uncover unanticipated interesting or meaningful patterns.

In this paper we explore simple scenarios in which heterogeneous but connected sensors are "over-deployed" in a room environment and start searching for correlations in their output. If the environment under observation is non-trivial, it is highly likely that comparing the data streams emanating from such a rich sensor network will uncover meaningful association previously unnoticed and often unsuspected. Although interesting in their own right, such associations might in fact have a significant practical value as they can contribute to the robustness of the sensing process. For instance, if it is observed that large readings from sensor A are usually accompanied by a strong signal from sensor B, then the firing of sensor B might add support to a less than convincing peak in the signal from A that would otherwise have been missed.

This straightforward observation motivates the approach explored in this paper. More precisely, we assume that we are observing a rich environment (e.g. a home or office) through a large collection of heterogeneous sensors linked to a high-speed local network (LAN). In addition, the sensors have access to computational resources for the purpose of digital signal processing, statistical analysis, database storage and retrieval, etc. These computational resources might come in various forms and shapes: simple processing can be taken care of by embedded software, while heavy duty number crunching can be shipped off to high-end servers attached to the network. To be sure, in most cases the sensors have been put in position with a specific purpose in mind (e.g. face-detection and -recognition) and are assumed to perform this job competently. But we are especially interested in what we might get *for free* by detecting (both temporal and spatial) associations or correlations. We will use the term *opportunistic sensing* to indicate situations in which a sensor network is picking up on patterns, trends, co-occurrences and the like that are the result of a serendipitous discovery but that turn out to be valuable in confirming or predicting other sensor events.

1.2. Examples of Opportunistic Sensing

The following examples of opportunistic sensing might further clarify the issues we want to address in this paper. The scenarios described below refer to sensors (such as cameras, microphones, motion-detectors, beam interruption detectors, etc.) deployed in a home or office environment.

- If a during the course of a day a face-recognition algorithm is able to identify a particular face and associate it with the brightly coloured shirt that person is wearing, then that shirt can be used to identify this person even if the face is not visible (e.g. turned away from the camera). This is clearly *opportunistic* as this will only work if the person is wearing an easily identifiable or eye-catching shirt, which might not often be the case. But *if* it is the case, then harnessing this information will substantially increase the robustness of the

[*] R.T. performed this work while at CWI.

identification process.

- Thinking along similar lines, imagine that most pieces of clothing come with a functional RFID tag and that at some locations the sensor network has RFID reader capabilities. If one of these locations is also equipped with a face recognizer, then it is possible to associate the RFID tag with that particular person and subsequently use the RFID signal to follow the person around the premisses. Again this is clearly opportunistic, as the next day that person might be wearing a different set of clothing, and therefore different tags. However, within the course of, say one day, the RFID signal drastically enhances the identification capabilities.

- If a network is able to process and identify auditory signals it might notice that a particular ringtone (of a mobile phone, say) is always associated with a specific voice answering it. This could then be used to improve the robustness of person identification based on auditory analysis (as it then becomes possible to use both voice and ringtone signatures to certify the presence of a person).

- Another interesting example found in [9] shows how the statistical analysis of correlations and lag-times for distributed motion sensors (in an office or home environment) can be used to map out the location topology.

1.3. Motivation

First and foremost this work is motivated by the need to make event recognition by sensor networks more robust and reliable. Humans (and other animals for that matter) have an – as yet – unparalleled capacity to (often unconsciously!) pick up on subtle sensorial clues and link them together into a highly accurate recognizer. If sensor networks could emulate this resourcefulness and flexibility then this would undoubtedly result in significant performance improvement.

Another reason for our interest in the problem can be found in the need to develop a *prediction-based attention mechanism*. This harks back the notion (eloquently expounded in [5]) that biological neural networks are constantly engaged in using their afferent sensor input to make (short-time) predictions of what is to be expected next. A bottom-up attention mechanism is activated whenever the actual events fail to meet these expectations. In this paper we try to implement some strategies that will support the constant updating of opportunistically derived models aimed at predicting various sensor output. Significant deviation between the predictions and the actual observations serve to arouse the dormant attention module.

As a final motivation we mention *co-training of sensor networks.* To explain what we mean by this, consider a room in which a network of sophisticated cameras has been installed, each equipped with expensive tracking software. Now imagine that in addition, this room is also equipped with a large collection of simple interrupt sensors, positioned at various random positions along the walls. People walking in the room will create a time-varying interrupt pattern. By observing this pattern and comparing it to person tracking trajectories generated by the cameras it might be possible for a neural network to learn which interrupt patterns correspond to which trajectories. Once this is learned, the expensive (and intrusive) cameras could be removed, and we can now rely on the signals generated by the cheap interrupt network.

1.4. SenseNets: Networks of Sensors and Computation Engines

In this paper we reserve the term *SenseNets* for a sensor network that enjoys the following characteristics:

- It comprises a collection of heterogeneous sensors that collectively cover at least two complementary sensing modalities;

- The sensing is typically *dense*: i.e. overlapping in space and modality);

- The sensors have access to each other's output and all required computation engines (digital signal processing, databases, statistical analysis, etc.) through a high-speed local network (LAN);

- In addition, we assume that this network is running *data exploration and modelling software* that allows it to detect both temporal and spatial correlations between the various output signals;

- Finally, the significance of events is determined in terms of some high-level goal (e.g. track the children in the house).

1.5. Previous and related work

Most work in sensor networks addresses event detection and recognition in a well-defined setting (see e.g. [3] and the references therein) and relatively few papers address the problem of data mining or opportunistic sensing. One related research initiative is the MetroSense Project [4] in which mobile sensors (e.g. attached to cars or bikes) are recruited on the fly to report on environmental conditions in the area they are passing through.

In the next sections we will describe two simple scenarios that we used to conduct some preliminary experiments to explore some of these issues.

2. SCENARIO 1: LEARNING A SENSOR ACTIVATION SEQUENCE

This scenario is set in a fitness club where a person entering a locker room needs to swipe a smartcard through a reader on the locker cabinet to get access to his or her personal items. The lockers are located opposite the (only) door in the room (see Fig. 1 for sketch of the layout), so the normal activity pattern is one in which a person will enter through the door, cross the room to walk up to the locker cabinet, swipe his card to get access to his personal belongings, and then walk out again through the same door. The room is monitored by a webcam (cam 1) attached to the wall opposite the door (i.e. next to the lockers). This camera runs motion detection software as well as a face detector. In addition there is a second camera (cam 2) also running motion detector with a relatively narrow observation beam aimed at the door but essentially orthogonal to that of the cam 1 (again see Fig. 1). Hence the area near the door is monitored by both motion detectors (as well as the face detector in cam 1) while other parts of the room are either outside the motion sensor range or only observed by one of them. Due to this layout, a person entering the room will set off (in quick succession) both motion detectors and the face detector. Each sensor event is time stamped and logged in a central database which is monitored by analysis and supervision software.

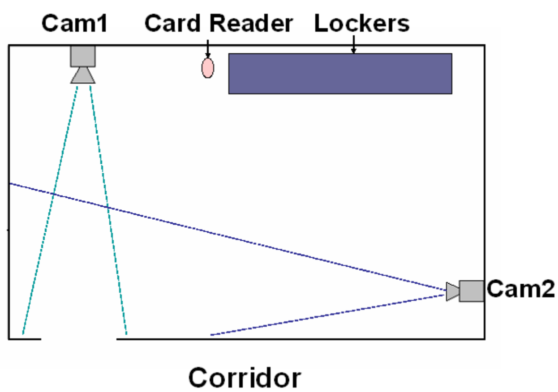

Fig. 1. Layout of locker room: camera 1 runs motion detection and face detection software, camera 2 is a visual motion detector.

For ease of visualisation, we represent the detection level of each sensor as a binary time series where the detection of motion or a face produces a spike of unit-length at the appropriate time. A typical example is shown in Fig. 3 where the bottom trace represents the detection events for motion detector 1 (in cam 1 facing the door), the trace above that shows the activity for motion detector 2 (in cam 2 grazing the door), the third trace responds whenever a face is detected (again by cam 1) and the top trace records a card swipe (the card's unique serial number is not recorded).

The system's "high-level goal" is to monitor access to the locker room. This can be achieved by detecting the swipes, but our aim is to search for (sequences of) other events that lead up to the swipe. Once we have spotted such correlations we can then use them as alternative detectors, or combine them to improve the robustness of the detection process.

Looking at the traces in Fig. 3 with these aims in mind, we notice that there is quite a lot of activity in the other sensors that seems uncorrelated to the swipes. This is clearly due to the intrinsic characteristics of the motion and face detectors. They can be set off by people walking by in the corridor or activities in other parts of the room. To get a better idea of the amount of association between these spike traces, we compute conditional probabilities as a measure for cross-correlation. More precisely, denote by $E(i,t) \equiv E_i(t)$ the binary sensor trace (as depicted in Fig. 3) for sensor i (with $i = 1,\ldots,4$ where 1 refers to motion detector 1, ..., and 4 to the card swipe) and time t runs through units $0, 1, \ldots, T = 1000$. We then compute for time-offset $\tau = -10, \ldots, 10$ the conditional probability of a spike (or event) for sensor j at time $t + \tau$ given that at time t a spike occurred for sensor i:

$$p_{ij}(\tau) = P(E(j, t + \tau) = 1 \,|\, E(i,t) = 1) \qquad (1)$$

Since τ takes on both positive and negative values, we look both forward and backward in time which means that we can detect possible "causes" as well as "consequences". The graphs for $p_{ij}(\tau)$ are displayed as a matrix in Fig. 4: The first row shows the conditional probabilities *given* that motion detector 1 (M1) fired at $\tau = 0$. The first element in this row shows the probability of other M1-events prior or after this event. The second plot shows that there is a slightly elevated probability that motion detector 2 (M2) will fire shortly afterwards, and the next plots in that row show how this effect ripples through (although attenuated) in the spiking activities of the face detector (3rd plot) and card reader (4th column). Since we're particulary interested in predicting a card swipe, we will focus on the fourth row which displays the conditional probabilities of sensor activity *given that a swipe has occurred at* $\tau = 0$. It clearly shows that a swipe is frequently preceded by a face detection (and to a lesser degree by spikes in the motion detectors). However, as transpires from the last graph on the third row (depicting probabilities conditional on a spike in the face detector) face detection in itself is *not* a good predictor of a subsequent spike ($p_{3,4}(\tau = 2) \approx 0.2$).

In this scenario, the challenge facing the *event analysis module* that is a standard part of the SenseNet is to figure out that while neither *face* nor *motion* detection in themselves are sufficient to predict a *swipe* event, a co-occurrence of all three is a strong predictor for a subsequent card swipe. So, if we think of the access to the lockers (as captured by the card reader) as a significant high-level/high-semantics events, then we are looking for the *low-level sensor correlates* to predict this high-level event. In the next section we explore various models.

2.1. Prediction models based on data analysis

2.1.1. Introduction

It is important to realise that the emphasis in this paper is on discovering associations the existence of which was not known. That means that initially we have no idea which "event states" are involved. For that reason we cannot simply apply standard HMM theory: we need to invoke exploratory data analysis first to uncover apparent underlying patterns. One such exploratory technique that has enjoyed a lot of attention recently is *Frequent Pattern Mining* (see e.g. Agrawal et.al. [1], or Joshi et.al.[6] for a survey). However, these techniques are difficult to apply if the order in which events occur is not fixed. For that reason we have taken recourse to more elementary — yet powerful — regression and prediction methodologies which we will discuss next.

2.1.2. Regression

Although the data are all binary (which means that tools from categorical data analysis such as logistic regression might be considered) it turns out that straightforward linear regression does a decent job in predicting the outcome. More precisely, since we are interested in prediction, we recast the event history time series (shown in Fig. 3) into a matrix by systematically selecting every time instance (t say) when a spike occurs in one of the four traces. Based on this we add a row to a predictor matrix X (with 30 columns) as well as to the response matrix y (single column) as follows: if the spike was produced by a swipe event, add 1 to (the bottom of) the y-column, otherwise add 0. At the same time, extract the values $E(i, t-10), \ldots, E(i, t-1)$ for each of the sensors $i = 1, 2, 3$ and concatenate them into a row of length 30 which is added to the bottom of X. Each row of X therefore encodes the spike activity in a short period immediately preceding the spike under consideration. The X and y data can now be fed into a linear regression model to estimate $\tilde{y} = Xb$ using stepwise regression. Since we are interested in binary events, the final estimate $\hat{y}$ is obtained by rounding $\tilde{y}$.

In a typical run, it turns out that stepwise regression consume between 10 to 12 variables of 30 ones available to it, and arrives at an optimal model that had an error-rate of approximately 3% with a confusion matrix for a total number of 518 recorded spikes:

	$\hat{y} = 0$	$\hat{y} = 1$
$y = 0$	461	6
$y = 1$	10	41

Optimality in this context means that the contribution of the remaining variables is not significant and would lower the adjusted R^2. (For completeness' sake we recall that $R^2_{adj} = 1 - (SS_{res}/df_{res})/(SS_{tot}/df_{tot})$). The history of inclusion

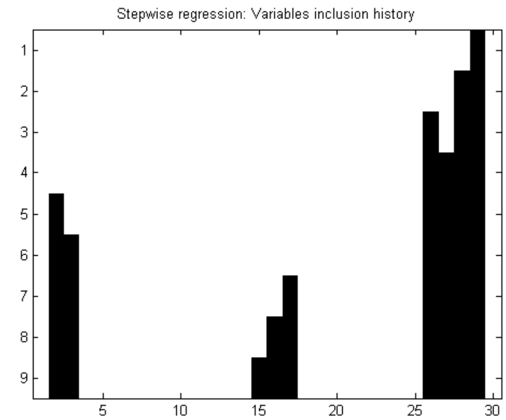

Fig. 2. Inclusion history for stepwise regression shows clusters around typical lag values. For more information see main text.

of the variables paints an interesting picture as can be seen in Fig. 2. The x-axis shows the 30 regression variables. Recall that variables 1 through 10 capture the spiking history in the first motion detector at times $(t-10, \ldots, t-1)$ where t is the moment when a spike occurred (in at least one of the detectors), variables 11 through 20 do the same for motion detector 2 as do 21 through 30 for the face detector. The black trails encode the successive entering of variables in the stepwise regression model, where the y-axis encodes the step number of inclusion. For instance, again referring to Fig. 2, variable 29 (i.e. whether or not a face was detected two time units before the occurrence of the spike under scrutiniy) was the first to enter, followed by variable 28, etc. The plot clearly shows that the first and most important variables to be included refer to the activity in the face detector in a short time period before the spike under scrutiny. Activity in the motion detectors at appropriately larger lag-times also makes significant contributions, which of course is in line with our understanding of the process. So based on these experiments we see that a simple regression model does a fairly good job at picking up the sequence of events that leads up to a card swipe.

2.1.3. Naive Bayes Classifier

As second modelling tool for the above classification/prediction task we tried a Naive Bayes (NB) classifier which also can be used to predict the next state based on the recent states of the sensors. For this the conditional probabilities of $P(E_i(t-\tau)|E_i(t))$ and $P(E_i(t-\tau)|\neg E_i(t))$ (τ ranges from -10 to -1) are required, which can be calculated by monitoring the sensors for some time. Also the prior probability $P(E_i)$ should be estimated from the sensor history (see Fig.4). Once these data have been gathered, a prediction can be made using Naive

Bayes and the recent sensor history:

$$e_i(1) = argmax_{e_i \in \{E_i, \neg E_i\}} P(e_i) \prod_{t=-10}^{-1} P(h_i(t)|e_i(t))$$

Since the time between subsequent events varies it is difficult to predict when exactly some event is going to happen. Though the classifier actually predicts the events reasonably well, the actual time at which an event will take place is very difficult, if not impossible, to predict.

	$\hat{y} = 0$	$\hat{y} = 1$
$y = 0$	4926	21
$y = 1$	23	20

Notice that the reason that the total number of events in this table is larger than in the previous one has to do with the fact that we looked at all time instance, not just the spikes.

From this concise comparison it transpires that the straightforward regression model outperforms the Naive Bayes classifier.

3. SCENARIO 2: ASSOCIATION IN THE AUDITORY DOMAIN

Although most surveillance applications focus on video, there is a growing interest in audio as it can be a rich source of complementary information (see e.g. [2] or [8] for more information). In the experiment described in this paper we integrated two audio recognition modules as part of the sensor network. The first one aims at recognizing voices, the second one is a ringtone classifier. The motivation for this choice is that whereas voices are often difficult to pick up and identify in a noisy room or office scenario, both the volume and audio signature of ringtones are more easily recognized. Moreover, many users have personalized their ringtones so that it becomes possible (at least in a relatively small group) to associate a ringtone with a person.

Both voice and ringtone classifiers have been implemented using a two-tier detector-recognition activation scheme. The audio stream is first divided into overlapping 1-sec windows and for each window we determine whether the more specialized voice or ringtone recognizer should be triggered (part 1). If the answer to this is affirmative, we trigger the corresponding recognition module (part 2). In the next subsections we look at this in a bit more detail.

3.1. Voice recognition

To detect a speech signal we use the fact that it is known that speech as a characteristic peak at the syllabic frequency of 4Hz. We therefore computed this peak by applying a *Mexican Hat* filter on the *Fast Fourier Transform* at this specific frequency. Then, based on the last 100 recorded peak values, we can compute an empirical distribution of this peak value that we can compare to known distributions for speech and non-speech, using ???.

If speech is detected, then the voice recognition module is launched. To enable recognition, we transform each 100 ms signal window using 13-dimensional *Mel Frequency Cepstral Coefficients* (MFCC) features. We then cluster each feature separately using SVM, and finally use the last 100 recorded clusters to build an **a posteriori** probability of a given speaker having spoken during the last second.

3.2. Ringtone recognition

It turns out that most ringtones can be efficiently modelled as a repeated sequence of fundamental well-distinguished frequencies. We therefore search for the most peaked frequencies applying a *Mexican Hat* filter on the whole spectrum (computed using *Fast Fourier Transform*) for each position of a 100ms sliding window and isolate energetic sequences. Auto-correlation analysis then informs us about the presence of repetitions in the signal – a feature that is quite characteristic for ringtones. If there are, this signal can be considered as a ring-tone recording and then compared to extracts from the database computing an adapted *Dynamic Time Warping* distance between corresponding sequences. To do so, we first build a matrix of distances between elements of Y, which is our test sequence, and elements of X, which in our case is the result of the auto-concatenation of a database sequence. Then, we will compute the shortest monotonic path in this matrix between any element from the first row to any element in the last one. The sum of all elements along this path gives us a distance between the unaligned sequences. If one of these distances is small enough, then the ring-tone is associated to the corresponding one in the database. Otherwise, it creates a new entry in the database.

Experiments on a small dataset of 10 persons with 10 different ringtones show that — not quite surprisingly — ringtones are much easier to detect and recognise than voices. More precisely, in natural speech experiments, only 40% of the voices could be correctly identified, while for ringtones we scored a 80% success rate. Hence the optimal recognition strategy turned out to be to use the ringtone to ascertain presence of a person, and to fall back on voice recognition only if no ringtone data were available. Hence, dynamically associating these two supports a significantly more robust recognition process.

4. CONCLUSIONS AND FUTURE WORK

In this paper we have argued that sensor networks in which multimodal sensors are connected to each other and computational devices capable of conducting statistical exploratory data analysis, offer the possibility of serendipitous and opportunistic sensing in which unanticipated associations are detected and used. To illustrate this point of view, we have outlined two simple scenarios in which this type of on the fly data-mining proved useful.

Acknowledgements The authors gratefully acknowledge partial support by FP6 Network of Excellence MUSCLE (FP6-507752), BASIS and BSIK-BRICKS (The Netherlands).

5. REFERENCES

[1] R. Agrawal, T. Imielinski, and A.N. Swami. Mining association rules between sets of items in large databases. In P. Buneman and S. Jajodia, editors, Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, vol. 22(2) of SIGMOD Record, pp. 207216. ACM Press, 1993.

[2] M. Cowling and R. Sitte: *Comparison of techniques for environmental sound recognition.* Pattern Recognition Letters, vol. 24, pp. 2895-2907, 2003.

[3] M. Cristani, J. Bicego and V. Murino. *Audio-video integration for background modelling.* In Proc. of ECCV2004, 2004, pp. 202-213.

[4] Shane B. Eisenman, Nicholas D. Lane, Emiliano Miluzzo, Ronald A. Peterson, Gahng-Seop Ahn, Andrew T. Campbell. *MetroSense Project: People-Centric Sensing at Scale.* WSW06 at SenSys06, October 31, 2006, Boulder, Colorado, USA.

[5] Jeff Hawkins. *On Intelligence.* Owl Books, July 14, 2005.

[6] Malesh Joshi, George Karypis, Vipin Kumar: *A Universal Formulation of Sequential Patterns.* University of Minnesota, Minneapolis, Dept. of Computer Science, Tech. Report 99-21.

[7] Kevin B. Korb and Ann E. Nicholson. *Bayesian Artificial Intelligence.* Chapman and Hall/CRC, 2005.

[8] V. Peltonen. *Computational auditory scene recognition.* MSc Thesis, Tampere University of Technology, Finland, 2001.

[9] C.R. Wren, D.C. Minnen, S.G. Rao: *Similarity-based analysis for large networks of ultra-low resolution sensors.* Pattern Recognition 39 (2006), 1918-1931.

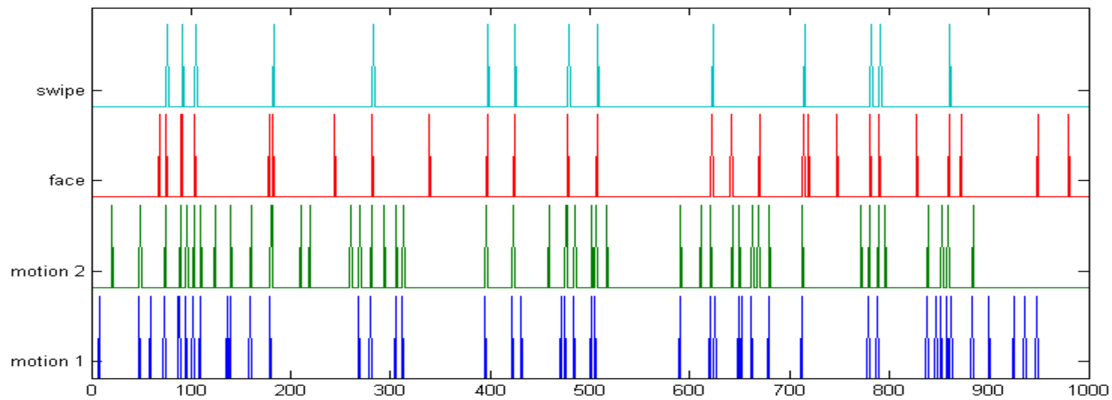

Fig. 3. Graphical representation of sensor activation (as a function of time) in the locker room scenario. Each spike represent a detection event by the corresponding sensor (top trace: card swipe, 2nd: face detection, 3rd and bottom trace: motion detectors.

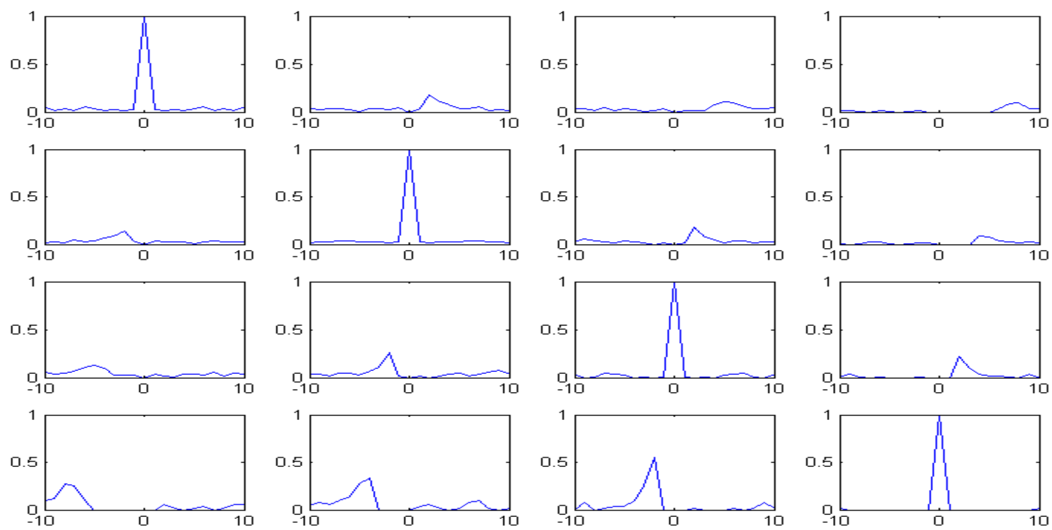

Fig. 4. Conditional probabilities for different events: the graph in the i^{th} row and j^{th} column displays the conditional probability of a spike in sensor j (as a function of the lag-time τ), *given that the* i^{th} *sensor has spiked at lag-time* $\tau = 0$. The i- and j-counter take on values 1 (motion detector 1), 2 (motion detector 2), 3 (face detection) and 4 (card swipe).