Detecting Contingency for HRI in Open-World Environments

Elaine Schaertl Short
University of Texas, Austin
Austin, Texas, USA
elaine.short@utexas.edu

Mai Lee Chang
University of Texas, Austin
Austin, Texas, USA
mlchang@utexas.edu

Andrea Thomaz
University of Texas, Austin
Austin, Texas, USA
athomaz@ece.utexas.edu

ABSTRACT

This paper presents a novel algorithm for detecting contingent reactions to robot behavior in noisy real-world environments with naive users. Prior work has established that one way to detect contingency is by calculating a difference metric between sensor data before and after a robot probe of the environment. Our algorithm, CIRCLE (Contingency for Interactive Real-time CLassification of Engagement) provides a new approach to calculating this difference and detecting contingency, improving the running time for the difference calculation from 2.5 seconds to approximately 0.001 seconds on an 1100-sample vector, and effectively enabling real-time detection of contingent events. We show accuracy comparable to the best offline results for detecting contingency in this way (89.5% vs 91% in prior work), and demonstrate the utility of the real-time contingency detection in a field study of a survey-administering robot in a noisy open-world environment with naïve users, showing that the robot can decrease the number of requests it makes (from 38 to 13) while more efficiently collecting survey responses (30% response rate rather than 26.3%).

CCS CONCEPTS

• **Human-centered computing → Empirical studies in HCI**; *Empirical studies in interaction design*; • **Computing methodologies → Intelligent agents**; • **Computer systems organization → Robotic autonomy**;

KEYWORDS

Human-Robot Interaction, Contingency Detection

ACM Reference format:
Elaine Schaertl Short, Mai Lee Chang, and Andrea Thomaz. 2018. Detecting Contingency for HRI in Open-World Environments. In *Proceedings of 2018 ACM/IEEE International Conference on Human-Robot Interaction, Chicago, IL, USA, March 5–8, 2018 (HRI '18),* 9 pages.
https://doi.org/10.1145/3171221.3171271

1 INTRODUCTION

Social robots deployed "in the wild", especially in open-world environments, may need to be able to model and understand the bi-directional relationship between their actuators and perceptual data in real time, without a priori information about what those changes may look like. Whether the robot looks around when a loud noise is heard, or waves and waits to see if anyone responds, these *contingent* relationships between robot behavior and the continuous numerical data of it's sensor inputs form one possible basis for rich, lively, and productive human-robot interaction (HRI).

(a) The robot greeting a member of the research team in the lab area. (b) The robot greeting a member of the research team in the public area.

Figure 1: The robot was deployed in two different locations: a quiet lab environment and a busy public area.

In this work, we operationalize *contingency* as a correlation between robot behavior and changes in the environment as measured by the robot's sensors. We focus on the case where the robot *tests* the environment by engaging in a behavior and waiting for a change. In this work we present a new variance-based real-time algorithm for detecting contingent changes in sensor data, CIRCLE (Contingency for Interactive Real-time CLassification of Engagement) and evaluate its performance and running time using in-lab acted data. We then test CIRCLE in an open-world deployment where it enables a robot to be appropriately proactive when conducting a survey.

We show that the CIRCLE algorithm results in runtime and accuracy improvements over a previous graph-based approach [17] from approximately quadratic to approximately linear time in the sample rate. In practice, this enables real-time use of the model: the running time for an 1100-sample vector, comparing an 8 second window of data before the robot's test to the 3 second window of data after, decreases from 2.5 seconds to approximately 0.001 seconds. We also demonstrate using ROC curves that the new approach has equal or greater discriminative power than the prior approach. We collect acted data in-lab and in the field, and live-annotate natural reactions to the robot in the field, and show that on these annotated datasets, CIRCLE gives 89.5% accuracy on lab data, 78.1% accuracy on acted data in the field, and 81.0%

HRI '18, March 5–8, 2018, Chicago, IL, USA

ACM ISBN 978-1-4503-4953-6/18/03…$15.00
https://doi.org/10.1145/3171221.3171271

accuracy on natural data in the field, giving similar results to the 91% accuracy of offline algorithm presented in [17] and improving on the 77% accuracy of the real time algorithm presented by [8].

We test the CIRCLE algorithm in a field study in the public area of an academic building, a high-noise and high-traffic environment, in a survey-collecting task. Using the CIRCLE algorithm, the robot is able to use short probes of the environment to choose appropriate times to make a longer speech requesting that people fill out a survey. We show that not only does CIRCLE match the accuracy of the cross-validation on training data when deployed in the field, but it also improves the robot's ability to engage with users. The robot using CIRCLE is more efficient at collecting responses by making less requests (made 13 requests and received 4 responses) than a baseline of the robot requesting at regular intervals (made 38 requests and received 10 responses), and spends less time speaking (13 requests in 2 hours rather than 38), thus causing less disruption.

2 RELATED WORK

Contingency is a fundamental characteristic of social human-human interaction and is an important dimension of adaptation throughout development [1, 6, 29]. As early as infancy, babies use contingency to detect responsive social agents [4, 30]. In addition, contingency learning allows people to obtain desirable outcomes or avoid unpleasant ones. For instance, a salesperson handing out flyers/sample products does not approach every person that walks by but instead uses various contingency cues to determine who to approach and initiate interaction. Thus contingency is one of the social intelligence skills that a robot must exhibit in order to be seamlessly integrated and accepted into society [2, 7, 31].

Not only must robots be able to detect contingent events, but they must do so in increasingly noisy and unpredictable environments, such as large public spaces. Tasks in these spaces may include advertising and providing route guidance to shoppers [14, 15], delivering snacks [18], distributing flyers to pedestrians [26], waiting and bartending in restaurants [3, 16], transporting goods and samples between departments in hospitals [20, 27], and more. These scenarios require the robot to understand and react to changes in the environment, especially human social behaviors. A part of this understanding includes being able to distinguish between contingent and non-contingent behavior, to predict who might be interested in interacting with the robot and who is unavailable or uninterested.

Most prior HRI research has also defined contingency as a change in a human interaction partner's behavior within a specific time window. They have investigated contingency by determining which mode(s) of communication and expected timing window contributes to a contingent response [5, 11]. Others have approached contingency as an engagement recognition problem. Yamaoka et al. [32] explored the effects of different levels of contingency with respect to the complexity of the robot's capabilities. Similarly, Pourmehr et al. [24] and Satake et al. [25] identified the most engaging person in a crowd for the robot to interact with and Nigam and Riek [23] classified whether it was appropriate for a robot to interrupt humans depending on the social context. We use a similar multi-modal approach, but provide a more general framework for detecting contingent responses.

Prior work in HRI on modeling contingency has used a single cue or combination of various cues including motion, body pose, audio, and gaze [8, 10, 17, 19, 22, 24, 28]. Most of this work was framed around the scenario that the robot initiates a contingent behavior and then detects if there is a contingent response from a human. Some of this work has focused on recognizing high-level features, such as body pose [24] and gaze [28]. However, these features can be computationally expensive and unreliable, and require the robot designer to correctly anticipate the range of responses to the robot that might occur. Using low-level features, Lee et al. [17] used an approach that detected a significant change between the time before and after the robot sends a signal/probe by generating graphs and analyzing the distance metric from the pre-probe samples to the distance metric from both the pre- and post-probe. A limitation of this approach is that it is computationally intensive and thus not feasible in a real-time interactive setting. Chu et al. [8] proposed an alternative by using a Support Vector Machine (SVM) classifier trained with positive and negative examples of contingency and achieved an average accuracy of 67% during real-time. Our work provides a generic approach that is feature-agnostic by modeling the changes in sensor data within a specific time window of the robot's behavior. Furthermore, the CIRCLE algorithm is computationally inexpensive, enabling the robot to detect unanticipated changes in noisy, high-frequency sensor data.

3 METHODOLOGY

Defining contingency as a correlation between robot behavior and sensor data, we present CIRCLE, a real-time algorithm for contingency detection over arbitrary sensor data that is parameterized using data from in-lab and in-the-field acted data, then evaluated in a field study in an open environment.

3.1 Model

As in prior work [8, 17], the basic approach to detecting contingency is for the robot to probe the environment with a behavior, then evaluate whether a change has occurred in some feature calculated from sensor data. To determine whether a change has occurred, the CIRCLE algorithm evaluates the relationship between the self-similarity of the signal during a period of time prior to a probe, and the cross-similarity of the signal after the probe to the signal before the probe. In order to enable real-time detection, we use ratio of the individual variances to the combined variance as a dissimilarity measure. This metric can be calculated in real time, in contrast to prior work that also calculated a change metrics, but with a graph-based model that could not be calculated in real time [17]. The change value is calculated as follows:

$$c(f_{pre}, f_{post}) = max(\sigma_{pre}/\sigma_{joint}, \sigma_{post}/\sigma_{joint}) \tag{1}$$

For multi-dimensional features, we calculate $\bar{c}$, the mean of the change values for each dimension.

CIRCLE (Algorithm 1) uses this change value in order to probe the environment and determine whether an event has occurred. The algorithm monitors the features at all times, and compares the change c_t after a probe at time t to the change values at each timestep within a window w steps into the past. If the change value is more than n standard deviations away from the mean of

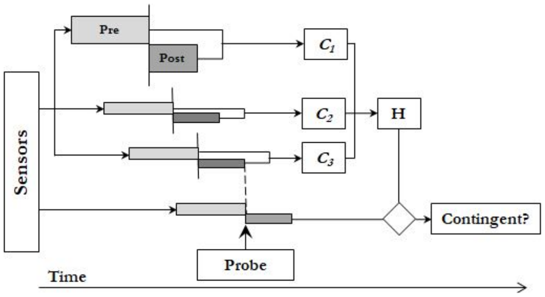


Figure 2: The approach of the CIRCLE algorithm: the changes in pre- and post-timestep data from multiple sensors (light and dark grey boxes) while the robot is not interacting with the environment is synthesized into a single comparison value (H), which is compared against the change in the data before and after the probe (bottom box).

Algorithm 1 CIRCLE: Contingency for Interactive Real-time CLassification of Engagement

```
for Each feature $f \in F$ do
    $H_t = \{\bar{c}(f_{(t-(\Delta_1+\Delta_2), t-\Delta_2)}, f_{(t-\Delta_2, t)}), t \in [t-(w+1), t-1]\}$
    Calculate $\mu_{0.2}(H_t)$, the 20% trimmed mean
    Calculate $\sigma_{0.2}(H_t)$, the 20% trimmed variance
end for
Engage in probe behavior
Wait for $\Delta_2$ seconds
$k_{test} = 0$
for Each feature $F$ do
    $c_{t'} = c(f_{(t'-(\Delta_1+\Delta_2), t'-\Delta_2)}, f_{(t'-\Delta_2, t')}$ (Equation 1)
    if $c_t > \mu_{0.2}(H_t) + n * \sigma_{0.2}(H_t)$ then
        Increment $k_{test}$
    end if
end for
Return $k_{test} > k$
```

the historical change values, then we consider an event to have occurred. The use of the change history allows the algorithm to adjust to periods of time with more changes in the features, such as during a class change in an academic building, when people pass into and out of the robot's view frequently. This approach is summarized in Figure 2.

In order to parameterize this algorithm for a specific application and environment, values must be chosen for the following:

- Δ_1 and Δ_2, the pre- and post-probe time windows
- w, the length of the history against which we compare (measured in minutes in this work)
- n, the number of standard deviations above the mean for which we consider a change to have occurred
- k, the number of features that need to be positive for a change to be counted as contingent

Additionally, features must be chosen that are informative relative to the types of contingent responses expected from humans in the environment. The value for Δ_2 is set to two seconds for the in-lab comparison to the prior work, which used a window of that length, but was set to one second for all subsequent iterations, based on results from the social science literature which indicate that a contingent response between people will occur within this time period [21]. For all other parameters, this work takes an iterative approach to selecting features and determining the parameter values, first using acted data collected in a lab environment, then acted data in the deployment environment, and finally using the behavior of naive users in the real-world environment.

3.2 In-Lab Evaluation with Acted Data

In order to evaluate potential features for inclusion in the field study, and to evaluate the running time of CIRCLE, we compared the ROC results to the highest-accuracy of the similar prior approaches: a graph-based approach that was evaluated on an in-lab dataset [17]. This prior approach took a bag-of-samples approach to comparing pre- and post-probe features, modeling the pre- and post-probe samples as graphs with edges corresponding to the distance between the multi-dimensional data points, and calculating a change metric based on the total distance between the nearest neighbors within and between the pre- and post-probe samples. In this section, we collect a similar dataset, and compare the prior approach to CIRCLE across features and sample sizes, using the area under the ROC curve of the change metric returned by the prior approach and CIRCLE.

Three actors who are members of our research team performed a set of ongoing actions in front of the robot, and either reacted (contingent) or ignored the robot (non-contingent) when it engaged in a probe behavior. The scenarios included: walking across the robot's visual field; walking across the robot's visual field while talking on the phone; standing in front of the robot while facing the robot; and standing in front of the robot while facing the robot and occupied with a task such as interacting with a phone.

This labeled data was used to evaluate potential features, including high-level features such as the position of objects and detected faces in the environment, low-level visual features, and audio features such as band energy and overall intensity. Using the area under ROC curves as we varied the thresholds on these features, we observed that the features most informative to the variance-based approach were those that could be calculated at a high frequency (at least 10-30Hz). From this set of features, a subset was selected that had good performance and could be calculated in real time: foreground object movement [13, 33, 34]; audio band energy in five bands, concentrated in the voice frequency range (<300Hz, 300-1200Hz, 1200-2100Hz, 2100-3000Hz, >3000Hz); spectral flatness of the audio signal; and audio intensity. Finally, because two of the three motion-based features were discarded, we added three additional low-level high-frequency motion features based on sparse optical flow [9]: average optical flow magnitude; the first difference of optical flow magnitude; and average optical flow angle.

The audio features were calculated for each of the four audio channels available through the Kinect sensor at 100Hz, and the visual features were calculated in each of four vertical segments in the robot's visual field. In addition to the segment averages, flow features were included representing the magnitude, first difference of magnitude, and angle of the vector sum of the optical flow of all

Feature	TPR	FPR	Acc.	n	w	Δ_1	Δ_2
Net OF (1st Diff.)	0.59	0.29	64.7%	1	5	2	4
Net OF (Angle)	0.47	0.18	64.7%	-0.3	5	8	3
Net OF (Magnitude)	0.65	0.24	70.6%	-0.3	10	8	1
Audio (Bands)	0.82	0.35	73.5%	-0.4	10	8	2
Audio (Intensity)	0.71	0.18	76.5%	0.8	5	2	1
Audio (Flatness)	0.53	0.18	67.6%	0.1	10	1	4
Motion	0.35	0.06	64.7%	-0.1	1	1	3
OF (Magnitude)	0.94	0.76	58.8%	-0.4	10	4	2
OF (1st Diff.)	0.65	0.47	58.8%	-0.1	5	1	2
OF (Angle)	0.53	0.18	67.6%	-0.5	1	8	4
All (untuned)	**0.65**	**0.0**	**82.4%**	**2.8**	**10**	**2**	**3**
All (tuned)	**0.82**	**0.12**	**85.3%**				

Table 1: Parameters for natural data, and their performance when trained and tested on the full dataset (14 positive and 14 negative examples, sampled from 14 positive and 116 negative examples). OF: Optical Flow.

Feature	TPR	FPR	Acc.	n	w	Δ_1	Δ_2
Net OF (1st Diff.)	0.615	0.385	61.5%	6	1	8	4
Net OF (Angle)	0.538	0	76.9%	-0.5	1	8	4
Net OF (Magnitude)	0.615	0.231	69.2%	-0.5	1	8	1
Audio (Bands)	0.923	0.615	65.4%	-0.5	1	1	2
Audio (Intensity)	0.846	0.538	65.4%	-0.1	5	4	2
Audio (Flatness)	0.615	0.231	69.2%	0	5	8	4
Motion	0.462	0.154	65.4%	-0.5	5	8	3
OF (Magnitude)	0.462	0.154	65.4%	-0.2	10	4	1
OF (1st Diff.)	0.615	0.231	69.2%	0	5	2	2
OF (Angle)	0.615	0.231	69.2%	-0.3	5	1	1
All (untuned)	**0.923**	**0.462**	**73.1%**	**0.1**	**1**	**2**	**2**
All (tuned)	**0.846**	**0.154**	**84.6%**				

Table 2: Parameters for acted data in the field, and their performance when trained and tested on the full dataset (13 positive and 13 negative examples). OF: Optical Flow.

Feature	TPR	FPR	Acc.	n	w	Δ_1	Δ_2
Net OF (1st Diff.)	1	0.643	67.9%	0.4	10	8	4
Net OF (Angle)	0.643	0.143	75.0%	0	5	8	4
Net OF (Magnitude)	0.714	0.357	67.9%	2	10	2	3
Audio (Bands)	0.714	0.143	78.6%	1.9	10	4	4
Audio (Intensity)	0.571	0.286	64.3%	5.6	5	8	2
Audio (Flatness)	0.714	0.214	75.0%	0.2	1	4	1
Motion	1	0.571	71.4%	-0.2	5	1	1
OF (Magnitude)	0.357	0.071	64.3%	0	10	4	1
OF (1st Diff.)	0.714	0.286	71.4%	1.6	10	1	3
OF (Angle)	0.714	0.214	75.0%	2.2	1	2	3
All (untuned)	**1**	**0.286**	**85.7%**	**1.6**	**10**	**8**	**3**
All (tuned)	**0.929**	**0.143**	**89.3%**				

Table 3: Parameters for acted data in the lab, and their performance when trained and tested on the full dataset (14 positive and 14 negative examples). OF: Optical Flow.

tracked points in the environment (in this feature, equal motion in opposite directions cancels out, thus measuring the *net* motion rather than the *average quantity of motion*).

The algorithm running time was calculated between the graph-based algorithm and the proposed algorithm was conducted with varying the sum of the pre- and post-probe windows from about 2 seconds total to about 11 seconds total, for each feature, as well as an additional feature composed of the concatenation of all of the features, sub-sampled to the audio feature rate of 100Hz. The change values obtained from the algorithms were compared using the area under the ROC curve (AUC) on a feature-by-feature basis.

3.3 Parameterization and Cross-Environment Comparison

With the final feature set, a matching set of examples were collected in-lab and in the field by one of the three previous actors, allowing for the comparison of parameter values and performance in the two environments, and to evaluate performance with realistic background noise and movement. A naturalistic dataset was collected by triggering the robot's probe behavior and coding the example as positive if any people in the robot's visual field reacted to the robot, and negative if no people reacted. Labels for the in-lab data were obtained from the behavior the actor was requested to produce (contingent or non-contingent); labels for the field data were obtained from a live coder observing in-person from nearby.

The deployment location was in the entry area of an academic building containing offices, classrooms, study areas, and a cafe. The robot was deployed near the high-traffic areas close to the entrance of the building, but not directly in the path of traffic. The number of people moving through the environment varies, with a comparatively high volume of foot traffic at the beginning and end of the business day and during class changes in the middle of the day, and a comparatively low volume of foot traffic early in the morning, during class times, and in the evening.

The actions were repeated in the lab and in the final deployment location with one of the three previous actors, using only the CIRCLE algorithm. Because of the improvement in running time, CIRCLE enabled us to compare 48 different combinations of parameters every second. Pre- and post- probe data were collected with $\Delta_1 \in \{1.0, 2.0, 4.0, 8.0\}$ and $\Delta_2 \in \{1.0, 2.0, 3.0, 4.0\}$ to enable evaluation of the algorithm performance relative to varying sizes of these windows. The values of $\mu_{0.2}(H_t)$ and $\sigma_{0.2}(H_t)$ were calculated with a a one-minute, five-minute, and ten-minute history window, and a time step of 1 second, for every combination of pre- and post-probe windows (48 different combinations per second). In order to simulate the real-world case where the robot would experience a mix of contingent and non-contingent behavior in the history window, the examples were collected at regular intervals over approximately two hours, alternating between positive and negative examples, meaning that the 10-minute history had on average one positive contingency example, and the 5-minute history had 0.5. The history collection was paused during the probes and calculated using a 20% trimmed mean to increase stability and reduce noise. A total of 28 examples were collected in lab and 26 examples in the field, balanced between positive and negative examples. The naturalistic data were collected over approximately 6 hours in the field, and included 14 positive and 116 negative examples, as most people passing through the space did not react to the robot.

The parameters were tuned by selecting the combination (out of 48 possibilities) of Δ_1, Δ_2, and w resulting in the largest area under the ROC curve with varying n. Then n was chosen to maximize the accuracy of the model (equivalent to maximizing TPR-FPR, since the datasets were balanced). In the tuned model, this process was repeated to maximize accuracy for each feature, then k was chosen to maximize overall accuracy, in the untuned model, this process was repeated for each possible value of k, and the values of k and n resulting in maximum accuracy were chosen. The algorithm performance was evaluated using 100 rounds of randomized 3-fold cross-validation, holding the proportion of positive to negative examples in the training and testing sets equal.

3.4 Parameterization

The parameterization of the models produced by the algorithm can be found in Tables 1, 2 and 3. These values were used to parameterize two versions of the model: a tuned version where the parameters were set for each feature individually, and an un-tuned version where the overall best set of parameters was chosen. The true positive rate (TPR), false positive rate (FPR), and overall accuracy for the given parameters when testing on the full dataset used to parameterize the model are also reported to provide a sense of the relative discriminative power of each feature on the full dataset. Cross-validated results are reported in Section 4.2. In the lab environment, the strongest feature is the audio band energies and audio flatness; in the field with acted data, the best individual feature is the net angle of sparse optical flow; and on the natural data, the best individual feature is the audio intensity. The values of w, Δ_1 and Δ_2 vary substantially for each feature from environment to environment; to test the performance of these parameters between environments, we evaluated the performance of the algorithm when parameterized in one environment and tested in another. As seen in Table 4, we found that there was poor transfer between environments, especially between the lab and field environments, with the lab-trained model mostly failing to label any in-the-field examples as positive and the field-trained models failing to label any in-the-lab examples as negative. There was better transfer between the acted and natural data in the field, but the natural data transferred better to the acted data than vice versa.

3.5 Field Study

In order to validate the use of the algorithm and verify the value of contingency detection to open-world HRI, we conducted a user study in the field location described in Section 3.3 over two days, in which the goal of the robot's behavior was to collect survey responses while causing minimal interruption to the people who use the space to study. The algorithm was tested on Poli, an approximately 1.5m tall mobile manipulation platform with a non-holonomic base, 6-degree of freedom arm, and pan-tilt head. The two embedded onboard computers were used for robot control and calculation of audio features, while an offboard computer[1], connected over a wired ethernet connection, was used to calculate visual features.

The robot was stationed in the same location as the data collections, alongside a table with a laptop running survey software. Based on the results of the cross-validation, the un-tuned parameter set from the natural in-the-field dataset was used (second-to-last line of Table 1): $k = 2, n = 2.8, w = 10, \Delta_1 = 2.0$ and $\Delta_2 = 3.0$. The algorithm was integrated into the robots behavior as follows:

(1) The robot probed the environment by waving and greeting.
(2) If a contingent event was detected using Algorithm 1, the robot made a request to complete a survey.
(3) The robot waited three minutes after making a survey request, to allow time for the survey to be completed.
(4) If no contingent event was detected, the robot stored the example and waited 45 seconds before probing again (resulting in a time between probes of about one minute).

This contingent behavior was compared to a baseline condition where the robot always made a survey request after the greeting. Each condition lasted for one hour and were counterbalanced. The sessions conducted on the first day was during the late afternoon and the next day's session was in the late morning. All of the sessions have one class change except the non-contingent condition on the second day that had two class changes.

To evaluate the efficiency of the contingency algorithm, we tracked the number of survey requests and the number of survey responses. The survey included questions about the robot's behavior, including the Perceived Sociability and Social Presence subscales of the UTAUT questionnaire [12] and two interaction-specific items on a 5 point Likert-type scale ranging from 1 to 5 (strongly disagree to strongly agree): *The robot knew when to respond to people.* and *The robot knew how to behave in this location.* However, we found that few users completed the entire survey, and so focused on whether or not the user started the survey as an outcome measure, rather than analyzing the specific responses. Instead, a sign was posted indicating that photography was permitted, and we measured the number of photos taken by users passing through the environment as a measure of engagement.

During all of the sessions, an observer from the research team observed the interactions from a nearby location and maintained a timestamped log of the robot's behavior (greeting and survey requests) and people's response to the robot. In the condition with the contingent robot, the live coder also recorded if the robot's decision to make a survey request is correct (i.e., number of true negatives, true positives, false negatives, and false positives). A contingent responses is defined as a change in the person's behavior such as the person stops his current task (e.g., walking, interacting with phone, talking to friends) to greet the robot, take a photo, and/or complete the survey.

[1] The offboard computer was used to enable monitoring of the interaction and to prevent version conflicts in libraries used in both the robot control stack and CIRCLE; all features used in this work could be calculated by the onboard computers alone.

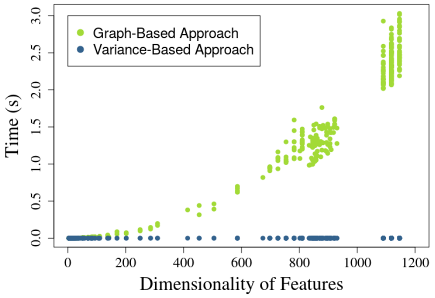


Figure 3: Comparison of running time

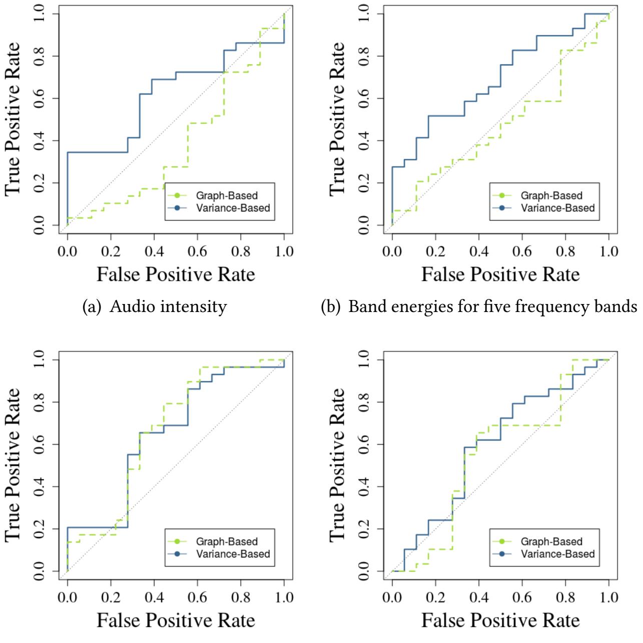
4 RESULTS

In this section, we present results for each of the three phases of this work: a comparison against prior work using in-lab acted data; testing accuracy on acted in-lab data, acted in-the-field data, and natural in-the-field data; and the results of the field study.

4.1 Algorithm Comparison on In-Lab Acted Data

The proposed CIRCLE algorithm has a substantial improvement in running time over the graph based approach of Lee et al. [17]. Intuitively, the graph based approach requires calculating the pairwise distances between every timepoint in the pre-probe and post-probe sensor data, giving it more-or-less quadratic running time in the dimensionality of the feature. The proposed algorithm, in contrast, uses the variance, which can be calculated in linear time on the dimensionality of the feature, and in fact, can be calculated online as new sensor samples are measured, although this latter improvement was not implemented in this work. Figure 3 shows the running time comparison of the two approaches, with the proposed metric shown in blue, based on the number of samples in the input (the product of the time-length of the sample and the sample size; the clusters represent different combinations of time windows and sample rates. In practice, CIRCLE can be calculated at more than 400Hz, for the largest feature size included in this data, which was an amalgamation of all other features and had a total size of around 1100 samples (8 seconds before the probe and 3 seconds after, at 100 Hz). The graph-based approach, in contrast, required up to 3 seconds to calculate the change metric on the same feature set.

Face and object tracking were not high-enough frequency to discriminate in most cases, so they were not analyzed. For the remaining features, AUC was calculated relative to an absolute threshold on the feature. The moving window history was not used due to the graph-based algorithm being too slow to calculate the history in real time. The variance-based approach performed better than the graph-based approach on three of the four remaining features and similarly on the fourth, with an AUC of 0.596 (versus 0.550) for the motion feature, an AUC of 0.655 (versus 0.669) for audio flatness, an AUC of 0.692 (versus 0.473) for band energies, and an AUC of 0.628 (versus 0.385) for audio intensity. The corresponding ROC curves can be seen in Figure 4.



(a) Audio intensity

(b) Band energies for five frequency bands

(c) Spectral flatness

(d) Pixel-level motion feature

Figure 4: Feature comparison, showing the new variance-based approach improving in discriminative power over the graph-based approach.

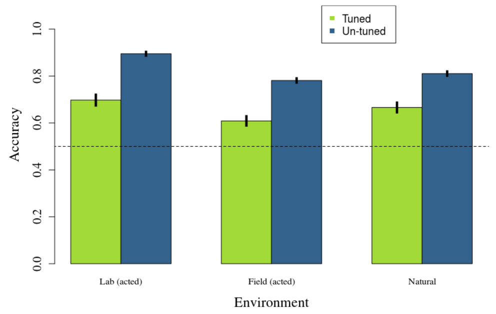


Figure 5: Accuracy after 100 rounds of randomly-sampled 3-fold cross-validation, sampling positive and negative examples separately to ensure equal proportions in the training and testing sets.

4.2 CIRCLE Algorithm Performance

In order to characterize the performance of our algorithm on the labeled data, we performed a cross-validation. The 3-fold cross validation was performed with pseudo-random sampling, with positive and negative examples sampled separately to ensure equal proportions in the training and testing sets. The cross-validation sample for negative examples in the natural dataset was further subsampled to create a balanced dataset. The results of 100 iterations of cross-validation are shown in Figures 5 and 6. The parameter sets that were not tuned on a per-feature basis performed better

	Lab	Field (Acted)	Field (Natural)
Lab	0.93/0.14	0.0/0.0	0.05/0.0
Field (Acted)	1.0/1.0	0.85/0.23	0.47/0.56
Field (Natural)	1.0/1.0	0.62/0.38	0.82/0.46

Table 4: Result of testing parameters trained on one environment in another (True Positive Rate/False Positive Rate). Row is training environment, column is testing environment. The full natural dataset of 116 negative and 14 positive examples was used for testing.

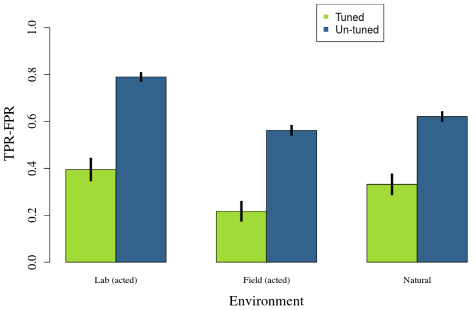


Figure 6: Difference between true positive and false positive rate after 100 rounds of randomly-sampled 3-fold cross-validation, sampling positive and negative examples separately to ensure equal proportions in the training and testing sets.

Day	Cond.	Probes	Requests	Submitted	%	Photos
1	C	28	6	1	16.7	2
1	NC	19	19	6	31.6	1
2	NC	19	19	4	21.1	3
2	C	37	7	3	42.9	5

Table 5: Results of the field test showing success rate of survey requests and photo sessions. Each condition session was one hour long. C: using CIRCLE; NC: not using CIRCLE.

Day	Cond.	TN	TP	FN	FP	Accuracy (%)
1	C	18	3	3	4	75.0
2	C	27	3	2	5	81.1

Table 6: Contingency detection field test results showing accuracy of CIRCLE algorithm. TN: True Negative; TP: True Positive; FN: False Negative; FP: False Positive.

in the cross validation: the lab data had a mean accuracy of 89.5% (*SD* = 3.9%), while the acted in-the-field data had a mean accuracy of 78.1% (*SD* = 4.5%) and the natural data had a mean accuracy of 81.0% (*SD* = 4.5%). In contrast, the tuned parameter sets had accuracy of 69.8% (*SD* = 11.4%) for lab data, 60.9% (*SD* = 9.8%) for acted in-the-field data, and 66.6% (*SD* = 10.3%) for natural data. The difference between the true positive and false positive rates followed the same pattern. For the un-tuned parameter sets these differences were 0.79 (*SD* = 0.078), 0.562 (*SD* = 0.089) and 0.621 (*SD* = 0.090) for the lab data, acted in-the-field data, and natural data, respectively. For the tuned parameter sets these values decreased to 0.395 (*SD* = 0.23) for lab data, 0.218 (*SD* = 0.20) for acted in-the-field data, and 0.332 (*SD* = 0.21) for natural data.

4.3 Field Study

Overall, the field results in Table 5 show that the contingent robot is better at collecting surveys. It made a total of 13 survey requests and received four submissions which is a success rate of 30.8% whereas in the non-contingent conditions, the success rate was 26.3%. In the Non-Contingent condition, the robot made requests after every probe where the majority were unsuccessful, whereas the contingent robot was much more constrained in only making requests when a likely responder was there and especially on the second day, this led to much higher success rate. We excluded two cases where a person walked up to the laptop and completed the survey without interacting with the robot first. We also did not analyzed the survey responses as the focus of the study was to evaluate the CIRCLE algorithm.

There were also times where a person was not within the robot's field of view but close enough to hear the robot's request so he came to interact with the robot and took the survey. Since the survey was in regards to the robot, people tried interacting in various ways with the robot as they assessed each of the questions. A solo person tended to be shy and stayed at a further distance from the robot whereas a group was more engaging and stood closer to the robot.

Moreover, the number of photo sessions are shown in Table 5. This included both single and group photos taken with the robot. If multiple people took photos at once, it was recorded as one photo. The contingent robot may be more engaging as shown with the higher number of photos taken on the second day. The low number of photos taken on the first day may have been due to the time of day that the study took place. It was late evening so people were probably tired and rushed to leave. Since the photos were recorded based on sessions, it is possible that multiple people were taking photos at the same time.

To understand how the CIRCLE algorithm performed, the human observer noted when the probe was sent and if the detection was correct. Table 6 shows the count of true negatives, true positives, false negatives, and false positives. The accuracy of the CIRCLE algorithm is 75.0% on the first day and 81.1% on the second day which is consistent with results from the lab data and acted data in the field. The algorithm performed well at detecting true negative scenarios which is important for this type of open-world environment where people are busy and moving fast. One of the challenges of this type of open-world environment is that it is characterized by periods of comparatively high and low traffic times. The contingent algorithm enabled the robot to probe the environment more often during the high traffic times, better handling the fast movement of people during those times. Conversely, during times of low traffic, the contingent robot's ability to reliably detect true negatives prevents it from making unnecessary requests.

5 DISCUSSION

In this work we developed an algorithm for detecting contingent changes in a robot's sensor data after a robot behavior. We validated the algorithm not only on acted in-lab data, but also in an open-world environment with high levels of background activity, including both noise and motion. Using a variance-based approach to detecting changes in noisy, computationally-inexpensive, low-level features, our algorithm brings together the high accuracy of a prior approach using a distance metric on a graph representation of the sensor data, with the real-time running time of a SVM-based prior approach. Overall, our algorithm can detect contingent changes in real-time with an accuracy of 89.5% on acted data in a quiet lab setting (comparable to the datasets used in prior work), 78.1% on acted data in the field, and 81.0% on natural data collected from naive users in the field and labeled online.

We also tested the parameters trained in one environment on the others, and found that there was poor transfer, especially between the lab and field environments, likely due to the substantial difference in the amount of noise in the two environments. The transfer from acted data in the field to natural data was slightly better, but there was better transfer from the natural data to acted data than vice versa. This suggests that it is necessary for researchers interested in detecting contingency in open-world environments to collect data in the field to get good accuracy. We demonstrate that our approach can be used across environments, including not only low-noise environments such as a lab, but also noisy open-world environments. To parameterize the algorithm, a balanced dataset of around 26-30 examples was needed, which could be collected in approximately 90 minutes for the acted data, and approximately twice as long for natural data, due to the strong bias towards negative examples. Future work will integrate this tuning process, by having a human in the loop for the first hours of deployment, then using the responses of users to the survey to autonomously obtain additional positive examples.

Lastly, we validated the CIRCLE algorithm by conducting a user study in the entry area of an academic building. Our results showed that the contingency detection algorithm enabled the robot to engage in requests to users that were timed to achieve a higher success rate for getting people to complete the surveys in comparison to the baseline. Additionally, the algorithm was successful at detecting true negatives in the contingent responses, avoiding bothering people in the environment by making requests when they were busy. Furthermore, this success was achieved without limiting the noise in the environment, or hand-coding a range of specific behaviors for the robot to detect, but by using noisy, low-level features to search for generic changes in the robot's environment. Our data show that most people passing through the environment are too busy or uninterested in interacting with the robot, demonstrating that this open-ended ability to detect relevant changes in the robot's sensor data is a critical skill for robots in the field.

The field study also demonstrated the need for the robot to both understand and produce contingent behavior. Natural interactions are continuous and ongoing, and after people responded to the robot's behavior, they expected the robot to respond in turn. Additionally, in some cases, the human would attempt to initiate the interaction, by waving or greeting the robot. We observed that these potential interaction partners would only wait a few seconds for the robot to respond before leaving. Future work will model the human's contingent experiences as well as the robots, to enable rich, ongoing interactions.

Our work does contingency detection without the use of external sensors; although this is more feasible than instrumenting the environment, and allows the algorithm to be deployed on a mobile robot, we found that the robot had a very short window for interaction: many people would pass through the robot's entire visual field in only 1-2 seconds. Because the robot was monitoring for changes in visual as well as audio features, people entering and leaving the robot's visual field were a major source of noise. Additionally, the history needed to be re-initialized whenever the robot turned its head. Future work might map the 2-D visual sensor data into the 3-D world to mitigate some of these issues. The timing of the robot's probes could also be improved: because probes were conducted at regular intervals, the robot might miss good opportunities to probe for contingency (when a human was in front of the robot and indicating interest). Future work will use this model of contingency not only to detect contingent reactions to the robot's behavior but also to inform the robot's behavior and ensure that humans in the environment have a contingent experience. A limitation of this work is that the labels were not able to be validated by a second coder, due to IRB restrictions against collecting video. However, the autonomous survey collection study demonstrates that the model obtained was useful, and future work will use a second live coder and synchronization software to obtain this validation. This work focused on the signal processing problem of contingency detection, and therefore limited the robot to simple behavior. Another direction for future work is to make use of the CIRCLE algorithm in the context of ongoing behavior, and to address the challenge of doing contingency detection on a mobile robot, where the baseline rate of visual and auditory change is high.

6 CONCLUSION

Contingency forms an important element of human-robot interaction; people expect that a robot will know when they are reacting to it, and that it will react in turn. Furthermore, due to its important role in the rhythm of interaction, contingency must be detected in real time. Our algorithm provides a means by which a robot can detect contingent reactions to its behavior with high accuracy in real time, not only in low-noise environments such as a lab, but also in high-noise environments such as public spaces. We demonstrate the utility of the CIRCLE algorithm in a field study in a public space, where humans in a noisy environment are completely unconstrained in their reactions to the robot, and show that using CIRCLE the robot is able to collect survey responses as effectively as if it were asking constantly, without the disruption to co-present humans that constant speech represents.

ACKNOWLEDGMENTS

The authors would like to thank Priyanka Khante and Alex Gutierrez for their assistance with data collection. This work was supported by NSF grant IIS-1724157.

REFERENCES

[1] Steven M. Alessandri, Margaret W. Sullivan, and Michael Lewis. 1990. Violation of expectancy and frustration in early infancy. *Developmental Psychology* 26, 5 (1990), 738–744. https://doi.org/10.1037/0012-1649.26.5.738

[2] Minoru Asada, Koh Hosoda, Yasuo Kuniyoshi, Hiroshi Ishiguro, Toshio Inui, Yuichiro Yoshikawa, Masaki Ogino, and Chisato Yoshida. 2009. Cognitive Developmental Robotics: A Survey. *IEEE Transactions on Autonomous Mental Development* 1, 1 (2009), 12–34. https://doi.org/10.1109/TAMD.2009.2021702

[3] M. Asif, M. Sabeel, and Mujeeb ur-Rahman an Z. H. Khan. 2016. Waiter Robot – Solution to Restaurant Automation. *Proceedings of the 1st Student Multi Disciplinary Research Conference (MDSRC)* November (2016).

[4] Ann E Bigelow and Philippe Rochat. 2006. Two-month-old Infants' Sensitivity to Social Contingency in Mother-Infant and Stranger-Infant Interaction. *Infancy* 9, 3 (2006), 313–325. https://doi.org/10.1207/s15327078in0903{_}3

[5] Nicholas J Butko and Javier R Movellan. 2010. Infomax Control of Eye Movements. *IEEE Transactions on Autonomous Mental Development* 2, 2 (2010), 91–107. https://doi.org/10.1109/TAMD.2010.2051029

[6] Richard L Canfield and Marshall M Haith. 1991. Young Infants' Visual Expectations for Symmetric and Asymmetric Stimulus Sequences. *Developmental Psychology* 27, 2 (1991), 198–208. https://doi.org/10.1037/0012-1649.27.2.198

[7] Angelo Cangelosi, Giorgio Metta, Gerhard Sagerer, Stefano Nolfi, Chrystopher Nehaniv, Kerstin Fischer, Jun Tani, Tony Belpaeme, Giulio Sandini, Francesco Nori, Luciano Fadiga, Britta Wrede, Katharina Rohlfing, Elio Tuci, Kerstin Dautenhahn, Joe Saunders, and Arne Zeschel. 2010. Integration of action and language knowledge: A roadmap for Developmental robotics. *IEEE Transactions on Autonomous Mental Development* 2, 3 (2010), 167–195. https://doi.org/10.1109/TAMD.2010.2053034

[8] Vivian Chu, Kalesha Bullard, and Andrea L. Thomaz. 2014. Multimodal real-time contingency detection for HRI. *IEEE International Conference on Intelligent Robots and Systems* Iros (2014), 3327–3332. https://doi.org/10.1109/IROS.2014.6943025

[9] Gunnar Farneb. 2003. Two-Frame Motion Estimation Based on Polynomial Expansion. *Lecture Notes in Computer Science* 2749, 1 (2003), 363–370. https://doi.org/10.1007/3-540-45103-X{_}50

[10] Markus Finke, Kheng Lee Koay, Kerstin Dautenhahn, Chrystopher L. Nehaniv, Michael L. Walters, and Joe Saunders. 2005. Hey, I'm over here - How can a robot attract people's attention? *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication* 2005 (2005), 7–12. https://doi.org/10.1109/ROMAN.2005.1513748

[11] Kevin Gold and Brian Scassellati. 2006. Learning acceptable windows of contingency. *Connection Science* 18, 2 (2006), 217–228. https://doi.org/10.1080/09540090600768435

[12] Marcel Heerink, Ben Kröse, Vanessa Evers, and Bob Wielinga. 2010. Assessing acceptance of assistive social agent technology by older adults: The almere model. *International Journal of Social Robotics* 2, 4 (2010), 361–375. https://doi.org/10.1007/s12369-010-0068-5

[13] P Kaewtrakulpong and R Bowden. 2001. An Improved Adaptive Background Mixture Model for Real- time Tracking with Shadow Detection. *Advanced Video Based Surveillance Systems* (2001), 1–5. https://doi.org/10.1.1.12.3705

[14] Takayuki Kanda, Masahiro Shiomi, Zenta Miyashita, Hiroshi Ishiguro, and Norihiro Hagita. 2009. An affective guide robot in a shopping mall. *Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction - HRI '09* (2009), 173. https://doi.org/10.1145/1514095.1514127

[15] Takayuki Kanda, Masahiro Shiomi, Zenta Miyashita, Hiroshi Ishiguro, and Norihiro Hagita. 2010. A communication robot in a shopping mall. *IEEE Transactions on Robotics* 26, 5 (2010), 897–913. https://doi.org/10.1109/TRO.2010.2062550

[16] Simon Keizer, Mary Ellen Foster, Zhuoran Wang, and Oliver Lemon. 2014. Machine Learning for Social Multiparty Human–Robot Interaction. *ACM Transactions on Interactive Intelligent Systems* 4, 3 (2014), 1–32. https://doi.org/10.1145/2600021

[17] Jinhan Lee, Crystal Chao, Aaron F. Bobick, and Andrea L. Thomaz. 2012. Multicue Contingency Detection. *International Journal of Social Robotics* 4, 2 (2012), 147–161. https://doi.org/10.1007/s12369-011-0136-5

[18] MK Lee and Jodi Forlizzi. 2009. Designing adaptive robotic services. *Proc. of IASDR'09* November (2009), 1–10.

[19] Timm Linder, Stefan Breuers, Bastian Leibe, and Kai O Arras. 2016. On multimodal people tracking from mobile platforms in very crowded and dynamic environments. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on.* IEEE, 5512–5519.

[20] Sara Ljungblad and Jirina Kotrbova. 2012. Hospital robot at work: something alien or an intelligent colleague? … *Cooperative Work* (2012), 177–186. https://doi.org/10.1145/2145204.2145233

[21] Javier R. Movellan. 2005. An infomax controller for real time detection of social contingency. *Proceedings of 2005 4th IEEE International Conference on Development and Learning* 2005 (2005), 19–24. https://doi.org/10.1109/DEVLRN.2005.1490937

[22] Steffen Muller, Sven Hellbach, Erik Schaffernicht, Antje Ober, Andrea Scheidig, and Horst Michael Gross. 2008. Whom to talk to? Estimating user interest from movement trajectories. *Proceedings of the 17th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN* (2008), 532–538. https://doi.org/10.1109/ROMAN.2008.4600721

[23] Aastha Nigam and Laurel D Riek. 2015. Social context perception for mobile robots. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on.* IEEE, 3621–3627.

[24] Shokoofeh Pourmehr, Jack Thomas, Jake Bruce, Jens Wawerla, and Richard Vaughan. 2017. Robust sensor fusion for finding HRI partners in a crowd. *Proceedings - IEEE International Conference on Robotics and Automation* i (2017), 3272–3278. https://doi.org/10.1109/ICRA.2017.7989373

[25] Satoru Satake, Takayuki Kanda, Dylan F Glas, Michita Imai, Hiroshi Ishiguro, and Norihiro Hagita. 2013. A robot that approaches pedestrians. *IEEE Transactions on Robotics* 29, 2 (2013), 508–524.

[26] Chao Shi, Masahiro Shiomi, Christian Smith, Takayuki Kanda, and Hiroshi Ishiguro. 2013. A Model of Distributional Handing Interaction for a Mobile Robot. *Robotics: Science and Systems* (2013). http://www.diva-portal.org/smash/record.jsf?pid=diva2:631957

[27] N.M.a Su, L.S.b Liu, and A.b Lazar. 2014. Mundanely miraculous: The robot in healthcare. *Proceedings of the NordiCHI 2014: The 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational* (2014), 391–400. https://doi.org/10.1145/2639189.2641216

[28] Hidenobu Sumioka, Yuichiro Yoshikawa, Masanori Morizono, and Minoru Asada. 2011. Socially Developmental Robot based on Self-Induced Contingency with Multi Latencies. *Proceedings of The 3rd International Conference on Cognitive Neurodynamics* (2011), 3–6.

[29] G M Tarabulsy, R Tessier, and a Kappas. 1996. Contingency detection and the contingent organization of behavior in interactions: implications for socioemotional development in infancy. *Psychological Bulletin* 120, 1 (1996), 25–41. https://doi.org/10.1037/0033-2909.120.1.25

[30] John S. Watson. 1972. Smiling, Cooing, and "The Game". *Merrill-Palmer Quarterly of Behavior and Development* 18, 4 (1972), 323–339. http://www.jstor.org/stable/23084026

[31] Mary-Anne Williams. 2012. Robot Social Intelligence. *International Conference on Social Robotics* 4 (2012), 45–55.

[32] Fumitaka Yamaoka, Takayuki Kanda, Hiroshi Ishiguro, and Norihiro Hagita. 2006. How contingent should a communication robot be? *Proceeding of the 1st ACM SIGCHI/SIGART Conference on Human-Robot Interaction - HRI '06* (2006), 313–320. https://doi.org/10.1145/1121241.1121294

[33] Zoran Zivkovic. 2004. Improved adaptive Gaussian mixture model for background subtraction. *Proceedings of the 17th International Conference on Pattern Recognition* 2, 2 (2004), 28–31. https://doi.org/10.1109/ICPR.2004.1333992

[34] Zoran Zivkovic and Ferdinand Van Der Heijden. 2006. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters* 27, 7 (2006), 773–780. https://doi.org/10.1016/j.patrec.2005.11.005