Big Data
Volume 4 Number 4, 2016

DOI: 10.1089/big.2016.0062

ORIGINAL ARTICLE

Detection of Subtle Context-Dependent Model Inaccuracies in High-Dimensional Robot Domains

Juan Pablo Mendoza,* Reid Simmons, and Manuela Veloso

Abstract

Autonomous robots often rely on models of their sensing and actions for intelligent decision making. However, when operating in unconstrained environments, the complexity of the world makes it infeasible to create models that are accurate in every situation. This article addresses the problem of using potentially large and high-dimensional sets of robot execution data to detect situations in which a robot model is inaccurate—that is, detecting context-dependent model inaccuracies in a high-dimensional context space. To find inaccuracies tractably, the robot conducts an informed search through low-dimensional projections of execution data to find parametric Regions of Inaccurate Modeling (RIMs). Empirical evidence from two robot domains shows that this approach significantly enhances the detection power of existing RIM-detection algorithms in high-dimensional spaces.

Keywords: anomaly detection; feature selection; robotics; robust autonomy

Introduction

Autonomous robots perform tasks by taking intelligent actions given potentially rich and multi-modal data from their sensors. One approach to achieve intelligent behavior is model-based decision making, in which the robot explicitly models the effects of its actions, and the meaning of its observations, to be able to plan how to perform its tasks effectively. Since the robot's actuators and sensors are subject to various forms of noise, these models are often stochastic.

Although ideally robot models would describe the stochastic dynamics of the world perfectly in every situation, this is often implausible for various reasons: It may be infeasible to collect training data from the entirety of the state space; the deployment environment may differ in unforeseeable ways from the training environment; or computational constraints may require the robot to use simple and efficient models. In these cases, the robot may have models that are generally accurate, but that fail to accurately represent the world dynamics in particular situations—that is, there are context-dependent model inaccuracies. Furthermore, these inaccuracies may be subtle, deviating only slightly from the robot's model; this subtlety requires the robot to analyze statistics of correlated sets of data, rather than individual data points, to find significant deviations from nominal execution. We note that this subtlety makes the problem intrinsically different from a binary classification problem: It is sets of correlated points, rather than individual points, that may be classified as anomalous.

Previous work has shown that detecting these subtle context-dependent model inaccuracies as parametric Regions of Inaccurate Modeling (RIMs) in the feature space of the robot can significantly improve the robustness of robot autonomy. In domains in which continuous execution is crucial, detection of RIMs online can significantly improve performance by applying appropriate model corrections.[1] In domains in which safe execution is crucial, detection of the RIMs can be supplemented by planning to avoid entering such potentially unsafe RIMs. In addition, reporting of these detected RIMs to human operators has led to discovery of algorithmic and modeling flaws in complex real-robot domains.[2]

Unlike previous work on RIM detection, this article focuses on high-dimensional robot domains. For complex robots with long-term deployment, as for other

School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania.

*Address correspondence to: *Juan Pablo Mendoza, School of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213,* E-mail: jpmendoza@ri.cmu.edu

Big Data domains, the ability to find patterns in large sets of high-dimensional data is crucial. In particular, complex robots often rely on a variety of sensors to collect rich, multi-modal execution data. Such rich data enable robots to maintain a large set of contextual features, such as estimated pose, velocity, battery voltage, time of day, presence of humans, and weather conditions, among many others. Since many of these features can be informative to the detection of RIMs, the scalability of RIM-detection approaches with domain dimensionality is crucial. This article presents an approach to RIM detection that significantly outperforms previous approaches as the dimensionality of the robot's feature space increases.

Illustrative example: golf robot

Figure 1 shows a simplified illustrative example of the type of problem that this article addresses. A robot shoots a golf ball from different locations on a field, obtaining a reward of 1 every time it hits the target on the right, and a reward of 0 otherwise.

Stochastic nominal model. During training, the robot uses a combination of domain knowledge and training data (Fig. 1b) to build a simple and generally accurate model of nominal behavior (Fig. 1c). If the robot always shoots in the direction of the target using a predefined behavior, its expected reward depends only on its position of the field: The robot expects a higher reward when shooting from closer to the target, since it is more likely to hit its target.

Subtle context-dependent model inaccuracy. During deployment—that is, after training—there is a bump on the field (Fig. 2a) that was not present during training and that is not directly perceptible to the robot's sensors. Because of this bump, the robot's execution differs from its model's predictions in a particular set of similar contexts: when the shot starts behind the bump (Fig. 2b, c). This model inaccuracy can be described as an RIM in the robot's context space, in which the collection of observations differ significantly from their modeled distribution. We note that each individual point, whether a success or a failure, does not contain enough information to detect an RIM.

High-dimensional domain. Similar to how the golfing robot's performance is affected by a bump on the field in this example, it could also be affected by various other

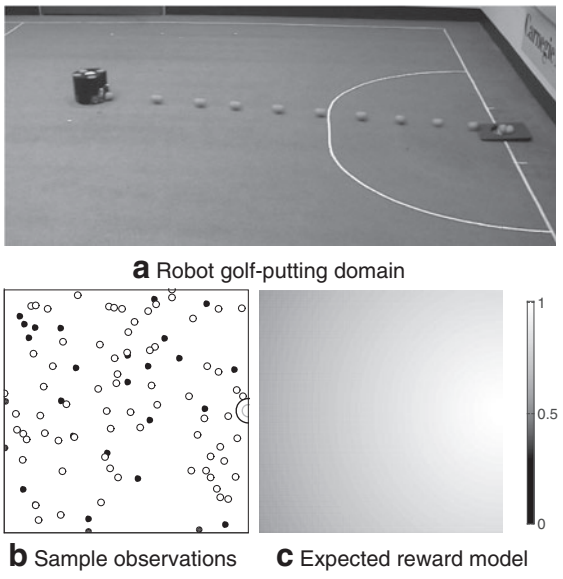


FIG. 1. **(a)** Nominal execution in the robot golf-putting domain. **(b)** Successful (white) and failed (black) shots on the target (black/gray concentric circles) follow the nominal expected reward model of **(c)**, as a function of shot location on the field.

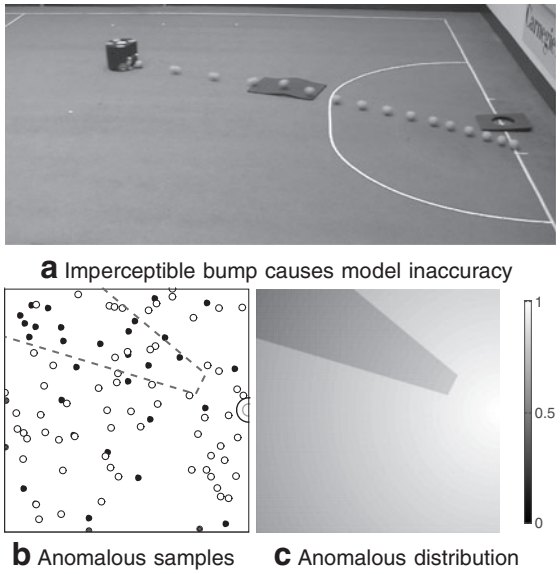


FIG. 2. **(a)** A bump on the field illustrates a context-dependent model inaccuracy. **(b)** The observations from behind the bump (dashed lines) differ significantly from the nominal expected reward distribution of Figure 1c, instead following the expected reward of **(c)**.

context variables, such as wind speed, lighting conditions, battery voltage, etc. Our approach must be able to find RIMs in this high-dimensional set of variables.

Contribution: RIM detection in high dimensions
This article contributes an approach for RIM detection that scales significantly better than existing approaches to high-dimensional domains.

Key assumption: low-dimensional RIMs. To find RIMs in high-dimensional context spaces efficiently and effectively, we assume that the RIMs can be fully described in a low-dimensional projection of the robot's observations, although the correct projection is unknown a priori. For example, the golfing robot's performance may be affected by an unseen bump on the field (2D RIM), by the wind velocity vector (3D RIM), or even by lighting conditions generated by the sun at a particular time of the day, during some months of the year in a particular section of the field (4D RIM); but not by an RIM that is intrinsically high dimensional. In practice, this assumption is met by many real-world model inaccuracies. The challenge of the problem, then, lies in identifying the best low-dimensional projection efficiently to find the RIMs in that subspace.

Feature selector for RIM detection. This article contributes a Feature Selector for RIM detection (FS-RIM) that scales well to high-dimensional and big data domains. The approach leverages previous work on RIM detection in low-dimensional spaces (e.g., Refs.[3–5]) to create a feature selector that alternatively conducts a heuristic best-first search for subsets of features that are likely to contain RIMs, and it explicitly searches for RIMs in the most promising projections, using existing low-dimensional approaches. We apply this algorithm to robotics domains, but its generality extends to other autonomous systems in which (i) the dimensionality of the domain is high, (ii) the system has a model of nominal behavior that is generally accurate, but (iii) the model may have inaccuracies in particular regions of context space. For example, previous work has applied RIM detection for early detection of disease[3] and homicide[6] hotspots in 2D maps to prevent epidemics. Our approach can enable such detection in context spaces well beyond two spatial dimensions on a map.

Evaluation and results. We evaluate the effectiveness of FS-RIM in two domains: simulated data from the golf-putting robot scenario above, and real motion data from the CoBot mobile service robots.[7] Results show that the robots are able to autonomously detect various types of injected model inaccuracies effectively and efficiently, significantly outperforming existing RIM-detection algorithms as domain dimensionality increases.

Paper organization. CoBot Mobile Robot Domain section describes the CoBot domain, highlighting its key properties: high-dimensional context, stochastic models of nominal behavior, and context-dependent model inaccuracies. The Background section contextualizes our problem and approach within existing execution monitoring and anomaly detection literature. The Feature Selection Algorithm section presents the technical details of our search-based feature selection algorithm, whereas the Empirical Evaluation section validates this approach through experiments on the golf-putting robot and CoBot domains. Finally, the Conclusion section concludes the article with a discussion of results and future work.

CoBot Mobile Robot Domain

The CoBots (Fig. 3) are mobile service robots that autonomously perform tasks for the inhabitants of the

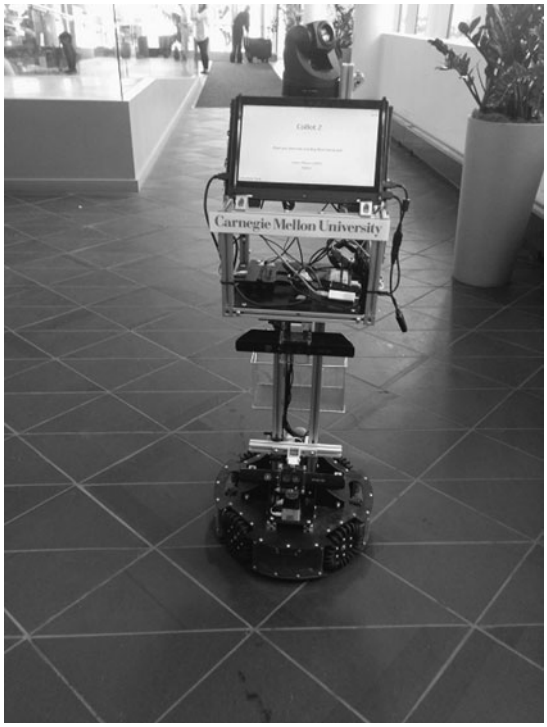


FIG. 3. CoBot mobile service robot.

Gates-Hillman Center at Carnegie Mellon University. Each CoBot has various stochastic models of nominal behavior that enable intelligent and robust performance, with more than 1000 km of autonomous navigation in the building[8]: A motion model translates desired robot translational and rotational velocities into motor currents; sensor models enable the robot to estimate its motion via wheel encoders and discern walls via laser rangefinders and depth cameras; task models enable the robot to estimate the time it will take to perform a particular task, and, thus, schedule various requested tasks accordingly.

Although the CoBot has generally accurate models for all of these, over its long deployment, it has encountered various context-dependent model inaccuracies that have negatively impacted its performance. For example, its motion is inaccurate when the robot moves at high speeds in a particular region of its building, such as bumps on the floor, corridors with rough terrain, or slanted ramps; the CoBot's depth camera is blinded when the robot goes to particular areas of its building, during particular times of the day, due to sunlight coming in through the windows; and its time to perform a task is significantly higher during times of the day when the corridors of the building are overpopulated, causing congestion. These are all examples of the context-dependent model inaccuracies that this work addresses; for this work, we focus on inaccuracies in the CoBot's motion model.

There is a high-dimensional space of context variables that may affect the robot's performance, such as its position, velocity and orientation, the presence of humans, the time of the day or day of the week, the presence of obstacles in its path, its battery voltage, and the amount of sun that shines into its sensors, among many others. This article examines model inaccuracies that affect different subsets of these features.

Projecting the data onto informative subsets of features can be crucial for finding RIMs of the CoBot's space. For example, Figure 4 shows data in which the CoBot's motion is subtly inaccurate in a particular corridor of the building; projecting the data onto the spatial location of the robot reveals a clear cluster of highly unlikely points in a particular corridor, likely to yield a region with high anomaly value, whereas projecting onto the angular velocity and time dimensions does not reveal any clear pattern of unlikely observations. On the other hand, Figure 5 shows data in which the CoBot's motion is subtly inaccurate when it turns left—that is, its angular velocity is greater than 0; projecting onto the angular velocity dimension reveals a cluster when the angular velocity is positive (the time dimension is shown simply for ease of visualization), whereas projecting onto the dimensions of the robot location does not reveal any clear pattern. We, thus, seek an approach that can reliably project the data onto informative dimensions efficiently.

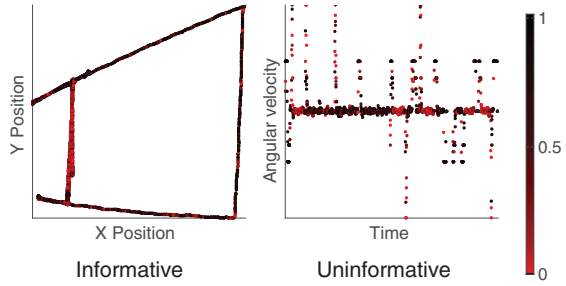


FIG. 4. Data from a motion inaccuracy affecting the CoBot in a particular corridor of its domain, shown in two different projections. The likelihood of each individual observation is shown by using the provided color scale.

Background

The problem of detecting and characterizing execution failures or anomalies has been studied extensively in robotics and other domains. This section contextualizes this work in relation to previous work in Execution Monitoring and Anomaly Detection.

Execution monitoring

The problem of execution monitoring, also called Fault Detection and Identification, or Diagnosis,[9] consists of

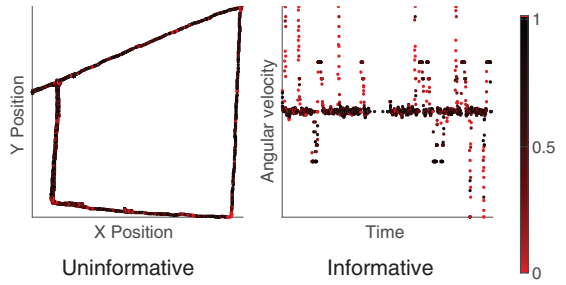


FIG. 5. Data from a motion inaccuracy that affects the CoBot only when it turns left, shown in two different projections.

detecting, identifying, and recovering from failures in execution. Execution monitoring is a well-established problem in various areas of scientific research, and the complex and stochastic nature of robotics domains has led to increased exploration of execution monitoring in the field.[10,11]

Execution monitoring can be divided into model-based methods, which monitor using models of the system, and model-free methods, which do so using only observed data.[12] This work focuses on domains in which robots have access to models of nominal execution; however, model-free methods have also been successfully applied in robotics.[13,14]

This work focuses on detecting contextual model inaccuracies given a list of stochastic contextual observations in which the robot's state $\boldsymbol{s}_t$ and action $\boldsymbol{a}_t$ can be mapped into a high-dimensional context feature point $\boldsymbol{x}_t$, and the outcome z_t is the observation to be monitored. Most work in execution monitoring has focused on fault detection by analyzing the likelihood of a single observation z_t or that of a sequence of observations $[z_i | i = 0, 1, \ldots, t]$—that is, using only time as a context that correlates various observations. Several algorithms have been developed to address this problem, and their properties, such as speed of detection and detection power, have been extensively studied.

Two of the most studied algorithms, sequential probability ratio test[15] and cumulative sum control charts,[16] detect faults by using thresholds on the likelihood ratio of residual observations, given a nominal model $\boldsymbol{\theta}^0$, and an alternative failure model $\boldsymbol{\theta}^1$. These algorithms can be very efficiently computed by maintaining an aggregate statistic S_t and updating it in constant time with a new observation $\mathbf{z}_t$. However, they require prior knowledge about the fault model $\boldsymbol{\theta}^1$, or sets of models $\{\boldsymbol{\theta}^i\}$ for multiple fault detection.[17] In contrast, we are interested in problems in which the anomalous distributions are not known a priori.

The generalized likelihood ratio approach[18] uses the maximum likelihood estimate of $\boldsymbol{\theta}^1$ for detection of faults with unknown parameters. Faults are again detected by thresholding a statistic S_t. However, in general domains, S_t requires computation that is not constant, but linear in t.

In our approach, we seek to use similar statistical techniques as these well-established methods. However, the key difference between our work and these and other time-series monitoring approaches[19–21] is that we need to detect inaccuracies that occur in particular regions of context space. Thus, we use methods that consider the full contextual observation $(\boldsymbol{x}_t, z_t)$, rather than just time t and the likelihood of the outcome z_t.

Anomaly detection

The anomaly detection community has extensively explored the problem of finding anomalies in contextual data. According to a classification proposed by previous survey work of anomaly detection research,[22] some of the important characteristics of our approach are: (i) it works on multi-dimensional continuous context data, (ii) it is semi-supervised—that is, it assumes that either a nominal model or nominal execution data is given, and (iii) apart from detecting anomalies, our approach also returns a measure of confidence on that detection [the anomaly value discussed later in Eq. (1)]. However, the main distinguishing characteristic of our work is that it can detect spatial collective anomalies—that is, anomalies that occur in particular regions of context space, and that require collections of data for detection, rather than individual points.

The problem of detecting spatial collective anomalies has received significant attention from the computer vision (CV) community, often for the different but related goal of image segmentation: This problem entails finding regions of an image that do not fit a model or their surroundings. Unfortunately, the algorithms developed for CV are not directly applicable to our problem, since they often rely on the structured and low-dimensional nature of images to detect anomalies (e.g., Refs.[23,24]).

Spatial scan statistics

The spatial scan statistic[3] is an approach for detecting regions of a multi-dimensional point process in which the number of observed points is significantly different from the number expected from a given model. This statistic has a wide range of applications, from forestry to astronomy[3]; however, it has been most often studied in the context of early disease outbreak detection. The core idea of the algorithm, given a set of contextual data $\boldsymbol{Z}$ and a model $\boldsymbol{\theta}^0$ of nominal behavior, is to search over a set of regions of context space to find the region R^* that maximizes the following log likelihood ratio:

$$\text{anom}(R, \mathbf{Z}, \boldsymbol{\theta}^0) = \ln \frac{P(\boldsymbol{Z} | \boldsymbol{\theta}^0 \text{is inaccurate in } R)}{P(\boldsymbol{Z} | \boldsymbol{\theta}^0 \text{is accurate in } R)} \quad (1)$$

More specifically, given a set $\boldsymbol{\Theta}$ of alternate possible models, and denoting by $\boldsymbol{Z}(R)$ the set of observations in $\boldsymbol{Z}$ that lie within R, Equation (1) becomes

$$\text{anom}(R, \mathbf{Z}, \boldsymbol{\theta}^0) = \ln \frac{\max_{\theta \in \Theta} P(\mathbf{Z}(R), \theta)}{P(\mathbf{Z}(R), \boldsymbol{\theta}^0)}. \quad (2)$$

This approach searches for the region R^* that is most likely to be anomalous, after which it conducts a statistical test to decide whether R^* is likely an anomalous region. Although the original exhaustive search algorithm[3] to find the most anomalous region R^* was sufficient for their two-dimensional search space, this approach does not scale well to higher-dimensional context spaces.

More recent work has extended the spatial scan statistics approach in several directions. A more efficient search algorithm has been proposed for axis-aligned rectangles,[25] but it does not scale well with the dimensionality of the domain. Graph-based approaches[6,26] are applicable in domains with a graph structure in their context space, such as border-connected regions of a map; however, these approaches are not applicable to our high-dimensional continuous-valued context spaces, which do not have an implicit structure. The Fast Subset Scan,[27,28] though efficient, limits its search to only regions of a fixed radius around each observation.

Focused Anomalous Region Optimization: RIM detection in low-dimensional domains

The Focused Anomalous Region Optimization (FARO) approach to RIM detection has been applied to online RIM detection[5] and recovery[1,2] in robotics domains. FARO has been shown to work well in domains of up to eight dimensions, but its performance degrades quickly with increasing dimensionality, as shown in the Empirical Evaluation section.

This article presents an approach for RIM detection in high-dimensional domains that acts as a wrapper around approaches for low-dimensional approaches such as FARO. Thus, for completeness, this section describes the FARO approach, presented in more detail in previous work.[5]

FARO anomaly value. FARO addresses the problem of finding the parametric region, out of a family of possible regions, that maximizes the anomaly value of Equation (2). In particular, this article focuses on model inaccuracies in which the observed mean of the distribution significantly deviates from the expected mean*. In this case, Equation (2) becomes:

$$\text{anom}(R, \mathbf{Z}, \boldsymbol{\theta}^0) = \ln \frac{\max_{\delta} \prod_{\boldsymbol{x}_i \in R} P(\boldsymbol{z}_i | \boldsymbol{\mu}(\boldsymbol{x}_i | \boldsymbol{\theta}^0) + \boldsymbol{\delta})}{\prod_{\boldsymbol{x}_i \in R} P(\boldsymbol{z}_i | \boldsymbol{\mu}(\boldsymbol{x}_i | \boldsymbol{\theta}^0))} \quad (3)$$

For the domains in this article, the robot models are given by normally distributed observations $\boldsymbol{P}(\boldsymbol{z}_i | \boldsymbol{x}_i, \boldsymbol{\theta}) \sim \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{x}_i | \boldsymbol{\theta}), \boldsymbol{\Sigma}(\boldsymbol{x}_i | \boldsymbol{\theta}))$. For brevity, when discussing the nominal model $\boldsymbol{\theta}^0$, we use the abbreviations $\boldsymbol{\mu}_i \equiv \boldsymbol{\mu}(\boldsymbol{x}_i | \boldsymbol{\theta}^0)$, $\boldsymbol{\Sigma}_i \equiv \boldsymbol{\Sigma}(\boldsymbol{x}_i | \boldsymbol{\theta}^0)$, $\Delta \boldsymbol{z}_i \equiv \boldsymbol{z}_i - \boldsymbol{\mu}_i$. anom $(R, \mathbf{Z}, \boldsymbol{\theta}^0)$ in Equation (3) becomes:

$$\begin{aligned}
&= \max_{\boldsymbol{\delta}} \sum_{\boldsymbol{x}_i \in R} \left[\ln \left(P(\boldsymbol{z}_i | \boldsymbol{\mu}_i + \boldsymbol{\delta}, \boldsymbol{\Sigma}_i) - \ln \left(P(\boldsymbol{z}_i | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \right) \right] \right. \\
&= \max_{\boldsymbol{\delta}} \sum_{\boldsymbol{x}_i \in R} \frac{1}{2} \left[\Delta \boldsymbol{z}_i^\top \boldsymbol{\Sigma}_i^{-1} \Delta \boldsymbol{z}_i - (\Delta \boldsymbol{z}_i - \boldsymbol{\delta})^\top \boldsymbol{\Sigma}_i^{-1} (\Delta \boldsymbol{z}_i - \boldsymbol{\delta}) \right] \\
&= \max_{\boldsymbol{\delta}} \sum_{\boldsymbol{x}_i \in R} \left[\boldsymbol{\delta}^\top \boldsymbol{\Sigma}_i^{-1} \Delta \boldsymbol{z}_i - \frac{1}{2} \boldsymbol{\delta}^\top \boldsymbol{\Sigma}_i^{-1} \boldsymbol{\delta} \right] \\
&= \max_{\boldsymbol{\delta}} \left[\boldsymbol{\delta}^\top \sum_{\boldsymbol{x}_i \in R} \left(\boldsymbol{\Sigma}_i^{-1} \Delta \boldsymbol{z}_i \right) - \frac{1}{2} \boldsymbol{\delta}^\top \sum_{\boldsymbol{x}_i \in R} \left(\boldsymbol{\Sigma}_i^{-1} \right) \boldsymbol{\delta} \right].
\end{aligned} \quad (4)$$

Substituting $\boldsymbol{\delta}_{\max}$ by its analytically derived value gives the expression for the quantity to maximize:

$$\text{anom}(R, \mathbf{Z}, \boldsymbol{\theta}^0) = \frac{1}{2} S_1^\top S_2^{-1} S_1 \quad (5)$$

$$S_1 \equiv \sum_{\boldsymbol{x}_i \in R} \boldsymbol{\Sigma}_i^{-1} \Delta \boldsymbol{z}_i$$

$$S_2 \equiv \sum_{\boldsymbol{x}_i \in R} \boldsymbol{\Sigma}_i^{-1}.$$

This is the value function that FARO optimizes, which depends only on two sufficient statistics S_1 and S_2 of the data contained in a region R.

FARO optimization space. FARO finds RIMs by optimizing over the parameter space of a particular family of regions. For the examples used in previous work and in this article, the search spaces for the optimization are ellipsoidal regions, which can express rotation, translation, and scaling, while requiring only $O(d^2)$ parameters. In a d-dimensional state space, an ellipsoid can be parameterized by a d-vector $\boldsymbol{u}$ and a $d \times d$ positive definite matrix $\boldsymbol{A}$ as the set of points that satisfy:

$$(\boldsymbol{x} - \boldsymbol{u})^\top \boldsymbol{A}^{-1} (\boldsymbol{x} - \boldsymbol{u}) < 1 \quad (6)$$

Thus, the parameter vector $\psi(\boldsymbol{u}, \boldsymbol{A})$ describing a particular ellipsoid is the linearized form of $\boldsymbol{u}$ and $\boldsymbol{A}$, consisting of $d + \frac{d(d+1)}{2} = \frac{1}{2}(d^2 + 3d)$ dimensions. The search space is then the space of such vectors $\psi \in \boldsymbol{\Psi}$ such that the matrix $\boldsymbol{A}$ is positive definite.

*Analogous mathematics can be used to detect model inaccuracies in which the variance of the distribution, and not the mean, is affected.

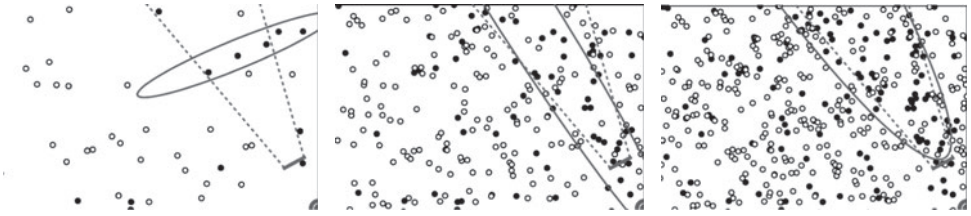


FIG. 6. Most anomalous ellipses found by FARO during three stages of execution. During nominal execution, the robot succeeds with probability 0.8. From behind the bump, it succeeds with probability 0.5. FARO, Focused Anomalous Region Optimization.

FARO optimization procedure. To find the ellipsoid that maximizes Equation (5), FARO relies on a nonlinear optimization method, such as the Cross-Entropy Method.[29] These methods are able to find the parameter vector ψ^* that maximizes the anomaly value $\text{anom}(R)$[†] of Equation (5). As an online method, FARO seeds the optimization with small regions surrounding the latest contextual observation of the robot each time it runs.

Figure 6 shows an example of the maximum anomaly regions detected by FARO in the simulated two-dimensional golf-putting domain. As the robot gathers more execution data, it is better able to outline the RIM generated by a bump on the field. However, as the Empirical Evaluation section shows, this remarkable performance degrades quickly in domains with higher dimensions.

Feature Selection Algorithm

This section presents the technical details of the main contribution of this article: an FS-RIM in high-dimensional domains. Similar to related work,[5] the goal is to find the region R^* of context space that maximizes the anomaly measure $anom(R)$ of Equation (2). The key assumption that enables FS-RIM to find R^* efficiently is that these RIMs are intrinsically low-dimensional regions that are embedded in a high-dimensional context space. This enables the use of a Feature Selection algorithm to greatly reduce the dimensionality of the search for RIMs.

In particular, FS-RIM is a wrapper-style Feature Selection algorithm,[30] in which the optimal projection of context space is found by evaluating the function to be maximized—that is, by finding R^*—in selected low-dimensional projections of the full context space. Choosing a wrapper approach enables the approach to leverage previously existing low-dimensional approaches (e.g., exhaustive search[3] or the FARO optimization-based search[5]) in its search for high-dimensional RIMs. Thus, the approach assumes that there exists a function, called $\text{findRIM}(\boldsymbol{Z}, \boldsymbol{\theta}^0)$, which, given a set of low-dimensional contextual observations $\boldsymbol{Z}$ and a model of nominal behavior $\boldsymbol{\theta}^0$, can find the region R^* that maximizes Equation (2).

The core problem is, thus, to efficiently and effectively search through the space of possible projections to explore more informative ones, such as those of Figures 4a and 5b, before exploring less informative ones, such as those of Figures 4b and 5a, since there exists $2^{|F|}$ different possible subsets of the full set of features F in the context space. FS-RIM uses an informed best-first search over elements $F \in 2^F$ of the power set 2^F of F. The search is conducted on a graph G in which the vertices $V(G)$ of the graph are the possible feature sets $F \in 2^F$, and edges $E(G)$ connect a vertex $v_0 = F$ to a vertex $v_1 = F \cup \{f\}$ if v_1 is the result of adding a single feature to v_0:

$$V(G) = 2^F \tag{7}$$

$$E(G) = \{(F, F \cup \{f\}) : F \in 2^F, f \in F, f \notin F\} \tag{8}$$

Figure 7 shows an abbreviated illustration of the first three levels of the resulting search tree, which always starts with the empty set of features as the root.

Algorithm 1 details the search procedure. The search starts by setting the boundary $\mathcal{O}$ of the search to the empty set of features (line 3). For each step i of the

[†]Since the observations $\boldsymbol{Z}$ and the nominal model $\boldsymbol{\theta}^0$ are constant throughout RIM-detection, $\text{anom}(R, \boldsymbol{Z}, \boldsymbol{\theta}^0)$ is abbreviated to $\text{anom}(R)$ when the arguments are clear, for brevity.

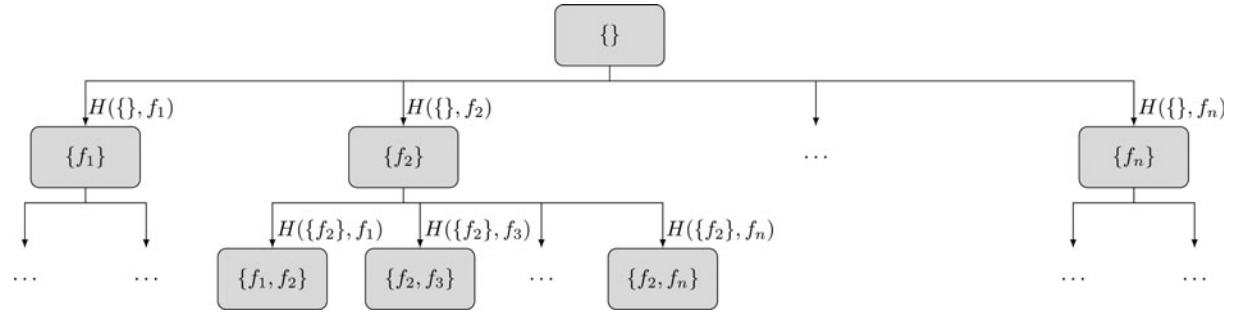


FIG. 7. Search tree for feature selection. The algorithm always starts the search with no features, and at each step searches the node on the boundary of unvisited nodes with the maximum heuristic value *H*.

search, the algorithm finds the edges that lie at the boundary of the search (line 5) and determines which one to explore by using a heuristic value function $H(e)$ (line 6). To explore the chosen edge (F_i^-, f_i), the algorithm first projects the original contextual observations $\boldsymbol{Z}$ onto the space spanned by the union $F_i = F_i^- \cup \{f_i\}$ (line 8). Then, the algorithm uses a low-dimensional search method findRIM to search for the most anomalous region R^* in the resulting low-dimensional space (line 9). Although this low-dimensional search method can be one of many options—for example, an exhaustive search if the low-dimensional space is small enough[3]—here, we use the FARO algorithm proposed by previous work.[5] Once the search has completed, and the most likely RIM R^* has been found, the algorithm decides whether the evidence is strong enough or not to declare a significant model inaccuracy (lines 14–18).

Algorithm parameters

Line 4 in Algorithm 1 specifies that the search continues until a domain-dependent maximum number of nodes $i_{\max}$ has been expanded. Depending on the requirements of the domain, this search-ending constraint may be exchanged by a time limit instead of a maximum number of expanded nodes. In general, the algorithm's performance may only improve with execution time or maximum number of nodes $i_{\max}$.

The threshold a_{thresh} in line 14 is domain specific, and it can be computed to correspond to a desired rate of false positive (FP) detections. As specified in previous work,[3] an approximate map from threshold to FP rate can be computed empirically through simulations of the domain under nominal execution—for example, to determine the anomaly threshold for an FP rate of 5%, one may run 100 simulations of the domain under nominal execution, run FS-RIM each time, and set the threshold a_{thresh} to the value of the fifth most anomalous region detected during nominal execution. In our experiments, we have used this 5% FP rate.

Algorithm 1. Algorithm for detection of an RIM in a high-dimensional context space.
Input: List of contextual observations $\boldsymbol{Z}$, nominal behavior model $\boldsymbol{\theta}^0$.
Output: An RIM R^*, or $\emptyset$ if no RIM is detected.

```
1: function FS-RIM (Z = [(x_0, z_0), ..., (x_n, z_n)], θ^0)
2:     R* ← ∅                          ▷ Most anomalous region thus far
3:     O ← {∅}                         ▷ Graph search boundary
4:     for i = [0, 1, ..., i_max] do
5:         E_i ← {(F ∈ O, f ∈ F) : f ∉ F}
6:         (F_i^-, f_i) ← arg max_{e∈E_i} [H(e)]
7:         F_i ← F_i^- ∪ {f_i}
8:         Z_i ← [(x_t^i, z_t) : x_t^i = Project(x_t, F_i)]
9:         R_i^* ← findRIM(Z_i, θ^0)
10:        if anom(R_i^*) > anom(R*) then
11:            R* ← R_i
12:        end if
13:    end for
14:    if anom(R*) > a_thresh(Z, θ^0) then
15:        return R*
16:    else
17:        return ∅
18:    end if
19: end function
```

Search heuristic

A crucial step in the Algorithm is the choice of heuristic function $H(e)$ in line 6. Computation of this heuristic must be relatively efficient in comparison to the findRIM function of line 9, since the former is invoked for each of the edges in the boundary of the search to decide which one is next explored with the latter. Furthermore, the heuristic must be as informative as possible for RIM detection—that is, it should approximate

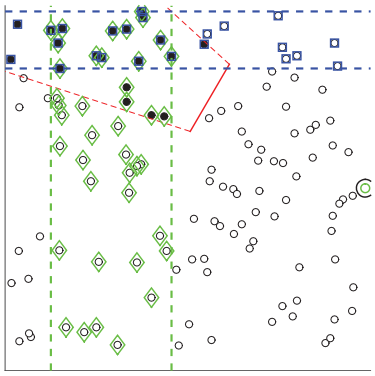


FIG. 8. Nonsutble golf domain RIM for heuristic visualization: every shot from within the RIM (red lines) is missed (black circles), whereas every shot from outside the RIM is scored (white circles). Blue dashed lines and squares show the maximum anomaly region when projecting onto $F=\{f_1\}$, whereas green dashed lines and diamonds show the maximum anomaly region along when projecting onto $f=f_2$. RIM, Regions of Inaccurate Modeling.

the anomaly value anom(R^*) of the maximum anomaly region R^* in that projection.

To compute efficient and informative heuristics for each edge (F,f), FS-RIM leverages the fact that the maximum anomaly region for the projections defined by F has been computed precisely in earlier stages of the search, and that the maximum anomaly region for all the individual features f can be computed only once in $O(|F|)$ time. Thus, the heuristics can use the corresponding maximum anomaly regions R^*_F and $R^*_{\{f\}}$ in their computations with little extra cost.

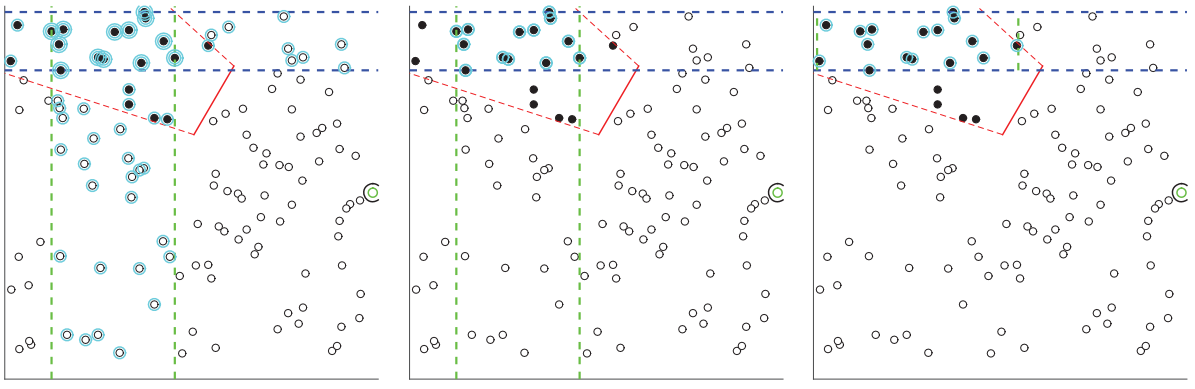
For visualization purposes, we describe each heuristic by using as an example an edge in which $F=\{f_1\}$ and $f=f_2$, shown in Figure 8. However, we note that F is a set that can contain zero or more features, whereas f is a single feature to be added to F. Thus, even though both R^*_F and $R^*_{\{f\}}$ appear as ranges along a single dimension in Figure 8, more generally, R^*_F is a $|F|$-dimensional parametric region (ellipsoid in this work), whereas $R^*_{\{f\}}$ is a one-dimensional parametric region. We note that neither $R^*_{\{f\}}$ nor R^*_F in Figure 8 contains all of the missed shots; this is because in their respective 1D projections, extending the region to contain every missed shot would also require containing many more scored shots, thus lowering the overall anomaly value *anom* of the region.

Anomaly sum heuristic (H_1). The first heuristic, computable in constant time, is given by the sum of the anomaly values of F and f:

$$H_1(e)=\text{anom}\left(R^*_F\right)+\text{anom}\left(R^*_{\{f\}}\right). \tag{9}$$

Figure 9a illustrates the meaning of this heuristic in the case where F is a one-element set: Heuristic H_1 is given by the sum of the anomaly values of each of the two regions independently.

For each feature f, the maximum anomaly region $R^*_{\{f\}}$ needs to be computed only once at the beginning



a Heuristic H_1 **b** Heuristic H_2 **c** Heuristic H_3

FIG. 9. Visualization of the proposed heuristics. Blue and green dashed lines show the relevant regions in lower dimensions, whereas cyan-highlighted data points are those that contribute to the heuristic value.

of the search, so its cost per edge of the search is constant. Furthermore, since the graph vertex containing F has already been explored, $\text{anom}(R^*_F)$ has already been computed by the time edge e is explored (see line 9 of Algorithm 1). This heuristic is very efficient, but may not be extremely informative in domains in which F and f may not seem highly anomalous independently, but they are together.

Region intersection heuristic (H_2). The second heuristic is more informative than H_1, but has a $O(n)$ computational cost per edge, where n is the number of data points. Given edge $e=(F,f)$, this heuristic is obtained by intersecting the points contained in the maximum anomaly region R^*_F from the subspace of features F, with those contained in the maximum anomaly region $R^*_{\{f\}}$ of the single-dimensional space of f, as illustrated in Figure 9b:

$$H_2(e)=\text{anom}\left(R^*_F\cap R^*_{\{f\}}\right). \tag{10}$$

In this work, where we use ellipsoids as our chosen parametric regions for optimization, this heuristic computes the anomaly value of the hyper-cylindrical region obtained from intersecting R^*_F and $R^*_{\{f\}}$: R^*_F forms the elliptical base in F, whereas $R^*_{\{f\}}$ constrains the points to those in a particular range along f. To compute H_2, each point in R^*_F is tested for belonging to the range $R^*_{\{f\}}$, leading to a $O(n)$ computation for each edge.

Conditional range heuristic (H_3). Finally, we present a highly informative heuristic with a $O(n^2)$ computational cost. Given edge $e=(F,f)$, H_3 computes the precise most anomalous range along dimension f, given only the observations contained in R^*_F, as illustrated in Figure 9c:

$$H_3(e)=\text{anom}\left(R^*_{\{f\}}|R^*_F\right). \tag{11}$$

Similar to H_2, this heuristic computes the anomaly value of a hyper-cylindrical region with base R^*_F. However, this region is the most anomalous such hyper-cylinder, and, thus, H_3 dominates H_2.

Heuristic H_3 can be computed in $O(n^2)$, because the most anomalous range $R^*_{\{f\}}$ along a single dimension f can be computed exactly in $O(n^2)$ by using dynamic programming, as explained in Appendix 1. The same procedure can be used to compute $R^*_{\{f\}}|R^*_F$, using only points within region R^*_F. In cases in which the number of points n is prohibitively high, computational costs can be diminished through the use of approximate methods for finding $R^*_{\{f\}}$ (e.g., Ref.[5]) or by binning points along feature f.

Empirical Evaluation

The performance of the contributed FS-RIM was evaluated via experiments on simulated data from the golf-putting domain and on real robot data from the CoBot domain. The primary purpose of this experimental validation is to demonstrate a significant performance improvement of RIM-detection algorithms in high-dimensional domains using FS-RIM, when compared with the FARO method without feature selection. In addition, the experiments provide a comparison among the different heuristic functions of the Search Heuristic section.

Evaluation metrics

The primary performance metric of FS-RIM is its ability to correctly identify data points that lie within an RIM. This is achieved by comparing each point's belonging to the ground truth RIM R^+ to its belonging to the maximum anomaly RIM R^* detected by FS-RIM, if any exists. For a given experiment, then, the number of true positives, FP, true negatives, and false negatives (FN) are given by:

$$\begin{aligned}
\text{TP} &= \sum_{(\boldsymbol{x}_i,z_i)\in\boldsymbol{Z}} \mathbf{1}(\boldsymbol{x}_i\in R^*\wedge \boldsymbol{x}_i\in R^+)\\
\text{FP} &= \sum_{(\boldsymbol{x}_i,z_i)\in\boldsymbol{Z}} \mathbf{1}(\boldsymbol{x}_i\in R^*\wedge \boldsymbol{x}_i\notin R^+)\\
\text{TN} &= \sum_{(\boldsymbol{x}_i,z_i)\in\boldsymbol{Z}} \mathbf{1}(\boldsymbol{x}_i\notin R^*\wedge \boldsymbol{x}_i\notin R^+)\\
\text{FN} &= \sum_{(\boldsymbol{x}_i,z_i)\in\boldsymbol{Z}} \mathbf{1}(\boldsymbol{x}_i\notin R^*\wedge \boldsymbol{x}_i\in R^+)
\end{aligned} \tag{12}$$

These measures are combined into a single standard F_1 performance metric, which evenly weights the precision and recall of the evaluated algorithms:

$$F_1=\frac{2\text{TP}}{2\text{TP}+\text{FP}+\text{FN}} \tag{13}$$

In particular, the performance of different detection algorithms is evaluated as a function of domain dimensionality. In particular, we hypothesize that the performance of FS-RIM would be comparable to FARO with no feature selection in lower-dimensional domains, but we expect to see a significant difference in higher-dimensional domains.

Furthermore, we evaluate the performance of FS-RIM compared with FARO as a function of computational running time. In both algorithms, the performance is expected to improve as the algorithm runs for longer. In FS-RIM, this is due to the search being able to expand more nodes; whereas in FARO, this is due to the optimization being able to explore more of the optimization

space. In both cases, we also expect the performance to plateau at a certain point in time, once the best option (locally best for FARO) has been found. Evaluating performance as a function of time is essential, because the best-first search method presented here has the goal of expanding nodes in an efficient order to avoid having to intractably search the entire space.

Golf-putting experiments

The first experimental domain is a variant of the golf-putting domain explained in the Introduction section. Although the binary-reward golf domain is useful for explanation and has been explored empirically in previous work,[5] here we explore empirically a continuous-reward variant: Instead of a binary reward of 0 or 1, the robot receives a continuous reward proportional to how close to the target the shot ends. This variant enables the work to focus on models defined as Gaussian distributions throughout; however, similar mathematical derivations can be used for other types of distributions.

Experimental setup.

Nominal behavior model. The golf-putting domain was set up as a highly controlled simulation, as an initial evaluation of FS-RIM with fully known ground truth. In this simulation, the locations p_i from which the robot shoots are chosen uniformly and randomly throughout the field. By design, the robot has a single action to shoot in the known direction of the target, and it receives a noisy reward r_i depending on p_i:

$$r_i = \bar{r}(p_i) + \epsilon \tag{14}$$

where $\epsilon \sim \mathcal{N}(0, 0.1^2)$ is a normally distributed noise term, and the expected reward $\bar{r}$ at location p_i is given by a linearly decreasing function of the distance d_i from p_i to the target: $\bar{r}(p_i) = 1.0 - 0.5\frac{d_i}{\text{FieldLength}}$. During training, the robot has access to the expected reward function $\bar{r}(p_i)$, and the parameters of the noise ε are extracted from the data.

Context-dependent model inaccuracy. A straight bump, such as that of Figure 2, is placed randomly on the field, in a different location for each experiment. The bump is always perpendicular to the target on the right, at a minimum distance of 5% of the field and a maximum distance of 75% of the field from the target. The center of the bump is placed at an angle between 135 deg and 235 deg from the target, and its subtended angle varies from $\frac{\pi}{8}$ to $\frac{\pi}{7}$ radians. When the robot takes a shot from a location p_i behind the bump, its expected reward is $\hat{r}(p_i) = \bar{r}(p_i) - 0.2$ instead of the original $\bar{r}(p_i)$, creating a region behind the bump in which the robot's model is inaccurate. For each of the experiments given later, the simulated robot repeatedly took shots until it had shot 30 times from behind the bump.

High-dimensional context. In addition to the two dimensions defining the spatial golf field dimensions, higher context dimensions were introduced as required for each experiment. The value of each data point in each of the added dimension is uniformly distributed in the range $[-1, +1]$. The desired outcome of this experiment, then, is for FS-RIM to be able to distinguish the two features that affect the model inaccuracy—that is, the x and y spatial dimensions—from the remaining dimensions, which are irrelevant to how well the model predicts the robot's reward.

Experimental results.

Figure 10 shows an example of the detected RIM in a 100-dimensional domain, using FS-RIM with heuristic H_3. The algorithm correctly identifies that the shown projection is the most informative one, and it proceeds to run an optimization over possible ellipses to find the one most likely to be an RIM.

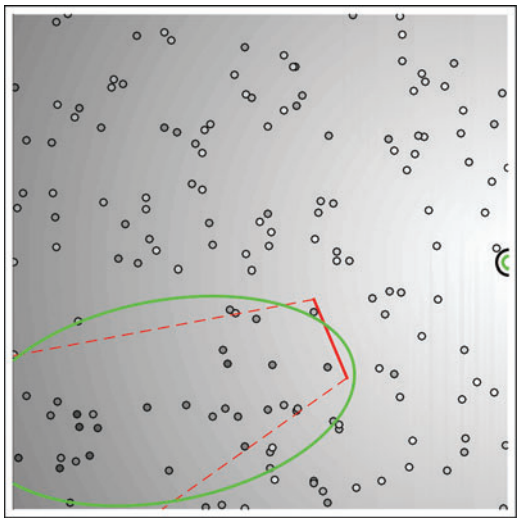


FIG. 10. Example of FS-RIM applied to a 100-dimensional golf domain. The green ellipse shows the detected RIM, whereas the red straight lines surround the ground truth RIM. The intensity of each point shows the received reward, whereas the background intensity shows the expected reward throughout.

Figure 11 shows the performance of FS-RIM using the three different heuristics of the Search Heuristic section (FS H_i), as well as that of the original FARO algorithm[5] without Feature Selection (No-FS). The first result is that the performance of No-FS quickly degrades with the increasing dimensionality of the domain. In a 2D environment, No-FS reaches peak performance before the FS methods, since it does not need to initially compute the heuristic values for each feature. However, this small-time advantage is overshadowed by the performance deficiency in higher dimensions.

Figure 11 also shows that the performance of the FS algorithms scales well with dimensionality. The time required to reach peak performance for each of the heuristics changes from about 18 seconds in the 2D domain, to between 20 and 30 seconds in the 100D domain. Furthermore, the performance of the algorithms, especially for heuristic H_3, does not degrade greatly between the 2D and the 100D domains.

CoBot experiments

The CoBot robots provide a platform to evaluate FS-RIM on real robot data. To run controlled experiments, the model inaccuracies were injected into the robot, during real execution, in precise regions of context space.

Experimental setup.

Nominal behavior model. The experiments in this article focus on the robots' motion models. The CoBot's motion commands are given by desired linear and angular velocities $v^d = [v_x^d, v_y^d, v_r^d]$, where v_x^d, v_y^d, and v_r^d are the desired speeds in the robot's forward direction, horizontal direction, and heading, respectively. These desired velocities are capped by known acceleration constraints of the robot, and they are transformed into individual wheel motion commands that a PID controller turns into the appropriate motor currents. As the robot moves, its wheel velocities are measured through wheel encoders and mapped back to robot velocity estimates $v^m = [v_x^m, v_y^m, v_r^m]$. After accounting for the latency of this process, a nominal model of wheel measurements is given by

$$\boldsymbol{v^m} = \boldsymbol{v^d} + \epsilon, \tag{15}$$

where $\epsilon \sim \mathcal{N}(\boldsymbol{\mu}_v, \Sigma_v)$ is the normal noise associated with robot sensors and actuators. The noise bias and variance parameters are estimated by using data from nominal robot execution.

High-dimensional context. The CoBot operates in an unconstrained office environment, and, thus, its domain is naturally high dimensional. To vary the dimensionality of the context space in these experiments, different subsets of the robot's context space were preselected:

7D context space: Time (1D), robot estimated position (2D) and orientation (1D), linear and angular velocity commands (3D).

15D context space: 7D context plus robot battery voltage (1D), progress information along the current navigation graph edge (3D), depth-camera plane-extraction statistics (4D).

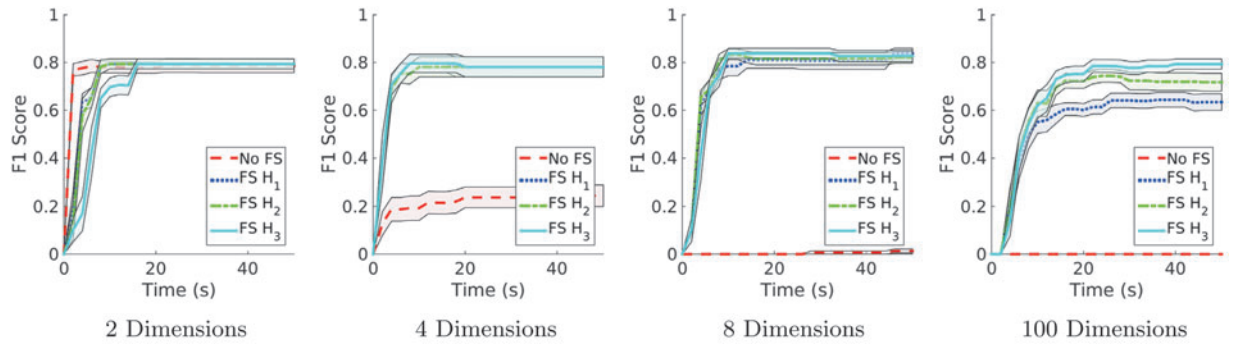


FIG. 11. Detection performance of FS-RIM with different heuristics (FS H_i) and no Feature Selection (No FS) as a function of algorithm running time, in the simulated golf-putting domain. Shaded areas show a standard error above and below the mean.

30D context space: 15D context plus depth values from 15 laser rangefinder rays, uniformly spaced along the rangefinder's field of view.

100D context space: 15D context plus depth values from 85 laser rangefinder rays, uniformly spaced along the rangefinder's field of view.

Context-dependent model inaccuracies. Two different types of model inaccuracies were injected into the robot's motion execution:

Corridor failure (Fig. 4): When moving in a particular corridor of the building, one of the robot's wheel encoders observes $0.95d$, at each timestep, where d is the displacement of the wheel observed during nominal execution. Thus, the RIM encompasses nonzero velocity points in a rectangular region of physical space.

Left turn failure (Fig. 5): The robot's execution is nominal except when it turns left (i.e., $\phi > 0$), in which case each of its wheels moves only at $0.95v$ for a velocity command v. Since the robot usually turns only at intersections or when it needs to face a doorway, this failure mode tests the algorithm when the anomalous data are quite far apart in physical space and time, but close along the angular velocity dimension.

These different types of inaccuracies affect different regions of the robot's context space, thus testing the generality of our feature selection algorithm.

Experimental procedure. For all the experiments, the CoBot was commanded, at a higher level, to navigate to various random points in the same floor as itself. The chosen path, as well as lower-level behaviors such as obstacle avoidance and localization, is handled by pre-existing algorithms.[31] The variance of the noise in Equation (15) was estimated from nominal execution data captured over $\sim$10 minutes of robot execution. Then, each of the testing anomalous conditions was run 10 times, each for $\sim$3 minutes of execution.

Experimental results. Figure 12 shows an example of a detected RIM on the Corridor Failure scenario. We note that the robot autonomously found the most informative projections for RIM detection, as well as the RIM approximation within the projected subspace.

Similarly, Figure 13 shows an example of FS-RIM in the Left Turn Failure scenario. In this case, FS-RIM finds the most anomalous region to lie only along the angular velocity dimension, and thus appears as the region above the horizontal green line in Figure 13a.

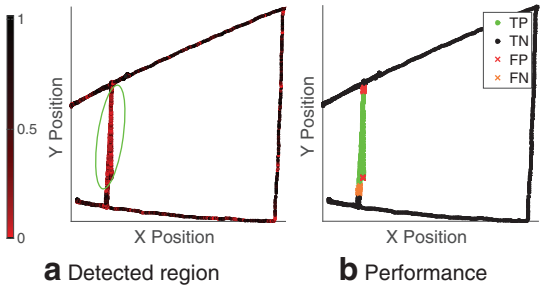


FIG. 12. Example of FS-RIM applied to the Corridor Failure. **(a)** An informative projection enables detection of a region containing collectively-highly-unlikely observations. **(b)** Detection performance.

Most of the FN in the Left Turn Failure scenario are points with angular velocity near 0. The deviation from nominal of these points is very small, since it is proportional to the angular velocity itself; thus, these points would not increase the anomaly value of the detected RIM.

Figures 14 and 15 show the performance of FS-RIM in the Corridor Failure and Left Turn Failure scenarios, respectively. Similar to the golf-putting results of the Experimental Results section, FS-RIM enables the robot to detect RIMs in high-dimensional domains much more effectively than not using FS-RIM. In these domains, the three heuristics did not show a significant

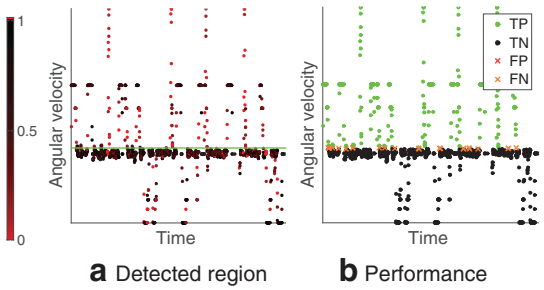


FIG. 13. Example of FS-RIM applied to the Left Turn Failure. **(a)** The optimal detected region lies only along the angular velocity dimension (above the green line). **(b)** Detection performance.

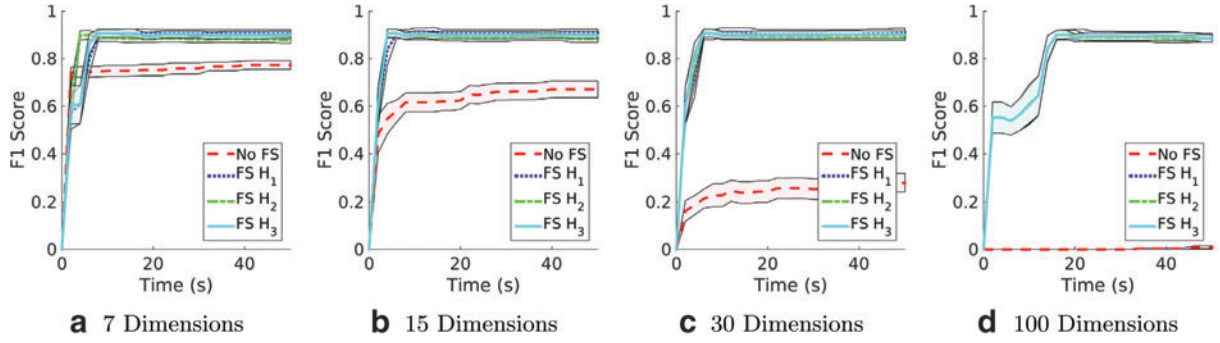


FIG. 14. Detection performance in the real-robot CoBot domain under a Corridor Failure. As the dimensionality of the domain increases by adding more features from execution, RIM detection using Feature Selection (FS H_i) with various heuristics significantly outperforms not using Feature Selection (No FS).

difference in performance from each other. We hypothesize that this, as well as the overall better performance of the various algorithms on these domains, is due to the higher density of data, which enables the robot to more clearly differentiate between nominal and anomalous execution. Figures 14d and 15d show a distinct shape: a quick increase in performance score, followed by a short plateau, followed by another increase and the final plateau. The first increase reflects the robot computing the anomaly value of the entire data set—that is, the 0D projection at the root of the search tree. The first plateau happens while the robot computes the maximum anomaly region $R^*_{\{f\}}$ for each feature f, as required for each of the heuristics of the Search Heuristic section. Finally, the next increase happens as the robot finds the right projection onto the 2D physical space in Figure 14d, and onto the 1D angular velocity space in Figure 15d.

Conclusion

This article presented the FS-RIM approach for scalable detection of RIMs in high-dimensional robot domains. Although the problem of RIM-detection has been addressed by previous work, we present an approach that scales well with the dimensionality of the robot's domain, provided that any anomalies can be characterized by a low-dimensional subspace of the high-dimensional context space of the robot.

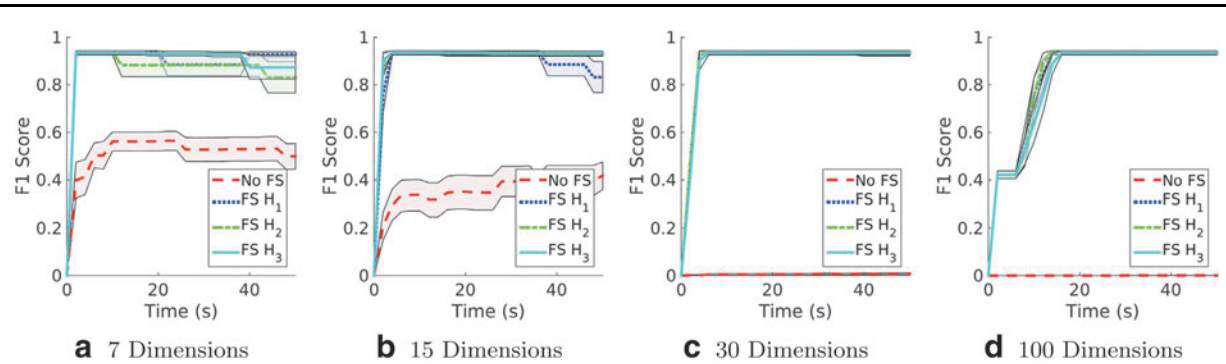


FIG. 15. Detection performance in the real-robot CoBot domain under a Left Turn Failure. As the dimensionality of the domain increases by adding more features from execution, RIM detection using Feature Selection (FS H_i) with various heuristics significantly outperforms not using Feature Selection (No FS).

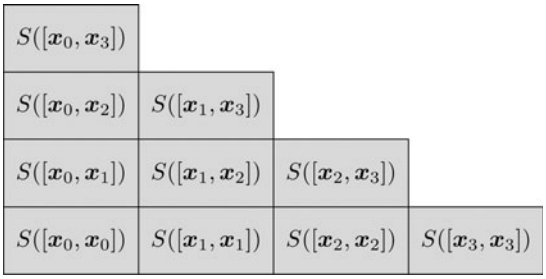


FIG. 16. Illustration of the dynamic programming procedure to obtain the range of maximum anomaly value in 1D. Each entry contains the sufficient statistics $S[\boldsymbol{x}_i, \boldsymbol{x}_{i+j}]$ required to compute $\text{anom}([\boldsymbol{x}_i, \boldsymbol{x}_{i+j}])$.

This approach frames the RIM-detection problem as an optimization problem in which the robot searches for the region R^* of its context space that maximizes an anomaly value anom(R). However, performing this optimization directly in the high-dimensional context space of the robot is a difficult problem. Therefore, FS-RIM instead optimizes in low-dimensional projections of the data. The selection order of these low-dimensional projections is conducted through a best-first-search process that uses various approximations of the anomaly value function as heuristics to guide the search.

Empirical evaluation of this detection approach was conducted in two robot domains: a fully controlled and an easily visualizable golf-putting simulation domain, and a semi-controlled real robot domain with injected motion anomalies. In both of these domains, the FS-RIM detector significantly outperformed existing approaches that do not use the Feature Selection as the dimensionality of the domain increases. Furthermore, since FS-RIM is a wrapper-style feature selector that uses a low-dimensional RIM detector at each step of its search, it may be used on top of various low-dimensional RIM detectors for high-dimensional RIM detection.

To the authors' knowledge, this is the first method designed to detect subtle context-dependent model inaccuracies in high-dimensional domains as RIMs. However, exploring comparisons to alternate possible approaches is an area of interest for future work. For example, one approach may include using decision trees[32] or random forests[33] to divide the space in a way that maximizes the anomaly value of one of the resulting regions. Although this approach would be best suited for axis-aligned RIMs, it could present other advantages over FS-RIM. Another example approach would be to use Gaussian Processes to create an approximation of the process distribution mean over the entire context space; however, such an approach may be prohibitively data intensive without some feature-selection preprocessing such as FS-RIM.

Although the FS-RIM detector has been evaluated exclusively in robotics domains, we expect its generality to extend well beyond to other domains requiring detection of anomalous regions in high-dimensional spaces, such as early detection of disease spread in rich data sets.

Acknowledgments

This material is based on work partially supported by NSF Grant IIS-1012733, DARPA Grant FA87501220291, and MURI subcontract 138803 of Award N00014-09-1-1031. The presentation reflects only the views of the authors. This material is based on research supported by (while Dr. R.S. was serving at) the National Science Foundation.

Author Disclosure Statement

No competing financial interests exist.

References

1. Mendoza JP, Veloso M, Simmons R. Plan execution monitoring through detection of unmet expectations about action outcomes. In: Proceedings of the International Conference on Robotics and Automation (ICRA), Seattle, May 2015, pp. 3247–3252.
2. Mendoza JP, Veloso M, Simmons R. Detecting and correcting model anomalies in subspaces of robot planning domains. In: Proceedings of International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), Istambul, Turkey, May 2015, pp. 1587–1595.
3. Kulldorff M. A spatial scan statistic. Commun Stat Theory Methods 1997;26:1481–1486.
4. Neill DB. Detection of Spatial and Spatio-Temporal Clusters. PhD Thesis. Pittsburgh, PA: Carnegie Mellon University, 2006.
5. Mendoza JP, Veloso M, Simmons R. Focused optimization for online detection of anomalous regions. In: Proceedings of the International Conference on Robotics and Automation (ICRA), Hong Kong, June 2014, pp. 3358–3363.
6. Duczmal L, Assuncao R. A simulated annealing strategy for the detection of arbitrarily shaped spatial clusters. Comput Stat Data Anal 2004;45:269–286.
7. Veloso M, Biswas J, Coltin B, Rosenthal S. CoBots: Robust symbiotic autonomous mobile service robots. In: Proceedings of IJCAI'15, the International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, July 2015.
8. Biswas J, Veloso M. The 1,000-km challenge: Insights and quantitative and qualitative results. IEEE Intell Syst 2016;31:86–96.
9. Cordier M-O, Dague P, Lvy F, et al. Conflicts versus analytical redundancy relations: A comparative analysis of the model based diagnosis approach from the artificial intelligence and automatic control perspectives. IEEE Trans Syst Man Cybern Part B 2004;34: 2163–2177.

10. Antonelli G. A survey of fault detection/tolerance strategies for AUVs and ROVs. In: Fault Diagnosis and Fault Tolerance for Mechatronic Systems: Recent Advances, Springer, 2003. pp. 109–127.
11. Pettersson O. Execution monitoring in robotics: A survey. Robot Auton Syst 2005;53:73–88.
12. Hwang I, Kim S, Kim Y, Seah CE. A survey of fault detection, isolation, and reconfiguration methods. IEEE Trans Control Syst Technol 2010;18:636–653.
13. Pettersson O, Karlsson L, Saffiotti A. Model-free execution monitoring in behavior-based mobile robotics. In: Proceedings of the International Conference on Advanced Robotics (ICAR), Coimbra, Portugal, June 2003. pp. 864–869.
14. Pettersson O, Karlsson L, Saffiotti A. Model-free execution monitoring by learning from simulation. In: Proceedings of the International Symposium on Computational Intelligence in Robotics and Automation (CIRA), Espoo, Finland, 2005. pp. 505–511.
15. Wald A. Sequential tests of statistical hypotheses. Ann Math Statist 1945;16:117–186.
16. Page ES. Continuous inspection schemes. Biometrika 1954;41:100–115.
17. Nikiforov IV. A generalized change detection problem. IEEE Trans Inf Theory 1995;41:171–187.
18. Willsky AS, Jones HL. A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems. IEEE Trans Autom Control. 1976;21:108–112.
19. Keogh E, Lonardi S, Chiu BY. Finding surprising patterns in a time series database in linear time and space. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002. p. 550.
20. Keogh E, Lin J. Hot sax: Efficiently finding the most unusual time series subsequence. In: Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM'05), 2005. pp. 226–233.
21. Bu Y, Leung OTW, Fu AWC, Keogh EJ. WAT: Finding top-K discords in time series database. In: Proceedings of 7th SIAM International Conference on Data Mining, April 2007, pp. 449–454.
22. Chandola V, Banerjee A, Kumar V. Anomaly detection: A survey. ACM Comput Surv 2009;41:1–72.
23. Felzenszwalb PF, Huttenlocher DP. Efficient graph-based image segmentation. Int J Comput Vision 2004;59:167–181.
24. Shi J, Malik J. Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell 2000;22:888–905.
25. Neill DB, Moore AW. Detecting significant multidimensional spatial clusters. Adv Neural Inf Process Syst 2004;17:969–976.
26. Tango T, Takahashi K. A flexibly shaped spatial scan statistic for detecting clusters. Int J Health Geogr. 2005;4:11.
27. Neill DB. Fast subset scan for spatial pattern detection. J Royal Stat Soc Ser B Stat Methodol, 2012;74:337–360.
28. Neill DB, McFowland E, Zheng H. Fast subset scan for multivariate event detection. Stat Med. 2013;32:2185–2208.
29. Rubinstein R. The cross-entropy method for combinatorial and continuous optimization. Method Comput Appl Probab 1999;1:127–190.
30. Kohavi R, John GH. Wrappers for feature subset selection. Artif Intell 1997;97:273–324.
31. Biswas J, Veloso MM. Episodic non-markov localization. Robot Auton Syst 2017;87:162–176.
32. Blockeel H, De Raedt L. Top-down induction of first-order logical decision trees. Artif Intell 1998;101:285–297.
33. Ho TK. The random subspace method for constructing decision forests. IEEE Trans Pattern Anal Mach Intell 1998;20:832–844.

Cite this article as: Mendoza JP, Simmons R, Veloso M (2016) Detection of subtle context-dependent model inaccuracies in high-dimensional robot domains. *Big Data* 4:4, 269–285, DOI: 10.1089/big.2016.0062.

Abbreviations Used

CV = computer vision
FARO = Focused Anomalous Region Optimization
FN = false negatives
FP = false positives
FS-RIM = feature selector for RIM-detection
RIMs = Regions of Inaccurate Modeling

Appendix 1. Computing R^* in 1D

In a one-dimensional domain, the convex region (e.g., ellipse) of maximum anomaly R^* can be computed in quadratic time, with respect to the number of data points, using dynamic programming. This algorithm exploits the fact that sufficient statistics for the anomaly measure of the union of two nonoverlapping regions R_1 and R_2 can be computed in constant time once the sufficient statistics for each of them has been computed. Thus, starting with regions that surround each individual data point, the maximum anomaly region R^* is found by merging adjacent regions and calculating their anomalies.

The anomaly measure of Equation (2) can often be computed from sufficient statistics of the data $\mathbf{Z}(R)$. For example, as shown in previous work,[5] when trying to find a shift in the mean of normally distributed observations, the logarithm of the anomaly $F(R) = \log \text{anom}(R)$ is computed as:

$$F(R) = \frac{1}{2}\left(\sum_{\boldsymbol{x}_i \in R} \boldsymbol{\Sigma}_i^{-1} \boldsymbol{\Delta z}_i\right)^{\top} \left(\sum_{\boldsymbol{x}_i \in R} \boldsymbol{\Sigma}_i^{-1}\right)^{-1} \left(\sum_{\boldsymbol{x}_i \in R} \boldsymbol{\Sigma}_i^{-1} \boldsymbol{\Delta z}_i\right) \tag{16}$$

where Σ_i is the expected covariance of observation z_i, and Δz_i is the deviation of observation z_i from its expected value, according to the nominal model. Thus, the statistics $S_{\Delta z} = \sum_{\boldsymbol{x}_i \in R} \Sigma_i^{-1} \Delta z_i$ and $S_\Sigma = \sum_{\boldsymbol{x}_i \in R} \Sigma_i^{-1}$ are sufficient for computing anom(R). Furthermore, given the statistics for two nonoverlapping regions $(S^1_{\Delta z}, S^1_\Sigma)$ and $(S^2_{\Delta z}, S^2_\Sigma)$, the statistics for the combination of their data is simply $S^{1+2}_{\Delta z} = S^1_{\Delta z} + S^2_{\Delta z}$ and $S^{1+2}_\Sigma = S^1_\Sigma + S^2_\Sigma$. Thus, it is possible to create a function anom(S) that computes the anomaly value from sufficient statistics of a region, and a

function merge (S^1, S^2) that merges sufficient statistics from two nonoverlapping regions; both of these run in constant time.

Given these sufficient statistics, Algorithm 2 describes the procedure of finding the region R^* of maximum anomaly in 1D. The algorithm finds the range $R^* = [\boldsymbol{x}_R^-, \boldsymbol{x}_R^+]$, where $\boldsymbol{x}_R^-, \boldsymbol{x}_R^+ \in \mathbb{R}$. First, the observations are sorted along their context dimension (line 2). This ordering enables the dynamic programming to create a table (line 3) such that, by the end of the procedure, this table $T[i][j]$ contains sufficient statistics of the observations in the range $\left[\boldsymbol{Z}'_j, \boldsymbol{Z}'_{j+i}\right]$ to compute its anomaly value. This is achieved by first storing the statistics of each individual point in $T[0][j]$ (lines 5–7), and then incrementally computing the statistics of larger regions by combining smaller ones (lines 8–12). Finally, the most anomalous range can be computed by finding the statistics in T that produce the maximum anomaly value (lines 13–15). Figure 16 illustrates the contents of the table for an example with four observations.

Algorithm 2. Algorithm to find the region R^* of maximum anomaly in a one-dimensional domain.
Input: Set of 1D contextual observations $\boldsymbol{Z}$, nominal model $\boldsymbol{\theta}^0$.
Output: The region R^* that maximizes $anom(R)$.

```
1: function FindAnom1D (Z = [(x_t, z_t)|t = 0, ..., N], θ^0)
2:     Z' ← sort(Z)
3:     ▷ 2D table stores stats of range [Z'_j, Z'_{j+i}]
4:     T ← table(|Z'|, |Z'|)
5:     for j ∈ |Z'| do
6:         T[0][j] ← anomStats(z_j)
7:     end for
8:     for i ← 1 to |Z'| do
9:         for j ← 0 to |Z'| − i do
10:            T[i][j] ← merge(T[i − 1][j], T[0][i + j])
11:        end for
12:    end for
13:    (i*, j*) ← arg max_(i,j) [anom(T[i][j])]
14:    R* ← [Z'_{j*} − ϵ, Z_{j*+i*} + ϵ]
15:    return R*
16: end function
```