Inference-Enabled Information-Theoretic Exploration of Continuous Action Spaces

Shi Bai, Jinkun Wang, Kevin Doherty and Brendan Englot

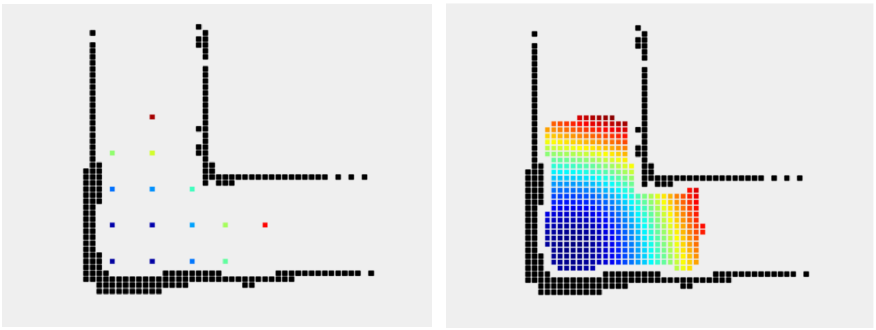
Abstract We consider an autonomous exploration problem in which a mobile robot is guided through an a priori unknown environment by a controller that chooses the most informative action within a local region. We propose a novel approach to efficiently evaluate information gain over the continuous action space that leverages supervised learning, with the anticipated mutual information achieved by a discrete set of action primitives serving as training data. We describe an autonomous exploration algorithm that uses this approach to cover a priori unknown environments. Computational results demonstrate that the method offers an improved rate of entropy reduction, surpassing a baseline approach that selects from the discrete action set, which in some instances requires more computational effort and yields less information.

1 Introduction

We consider a mobile robot that has no prior knowledge of the contents of its environment and must make repeated decisions about where to travel next, comprising an autonomous exploration problem [1]. Specifically, we formulate an information-theoretic exploration problem in which the long-term goal is to reduce entropy throughout the robot's environment map, and the short-term goal is to perform the sensing action in each iteration that will maximize mutual information, along the lines of [2]. We assume the robot is equipped with a range sensor and uses an occupancy grid [3] to represent and reason about the environment.

Motivated by the recent work of [4], which proved that a controller driven by mutual information maximization attracts a robot to unexplored space, our aim is to

Shi Bai, Jinkun Wang, Brendan Englot - Department of Mechanical Engineering,
Kevin Doherty - Department of Electrical and Computer Engineering,
Stevens Institute of Technology, Castle Point on Hudson, Hoboken, NJ, 07030.
e-mail: {sbai1, jwang92, kdoherty, benglot}@stevens.edu



(a) MI of discrete sensing actions.

(b) MI predicted by GP regression.

Fig. 1: An illustration of two steps of the proposed decision-making process. In (a), the current occupancy map is used to predict the mutual information at a set of discrete sensing locations. In (b), a Gaussian process (GP) regression is performed over a local region of the space of sensing actions, using the data from (a) as training data. The darkest of the blue cells in (a) represents the current location of the robot, and the black cells represent known obstacles in the robot's occupancy grid. The color scale from blue to red indicates increasing mutual information (MI).

implement a mutual information maximization approach that is amenable to real-time decision making and scalable to higher-dimensional systems. The approach of [4], although successful, requires a predictive evaluation of the mutual information achieved by performing every possible sensing action within a robot's finely discretized action space. We hope to cut down the complexity by evaluating only a select number of actions, and using these as training data for a supervised learning procedure that will predict information gain throughout the continuous action space.

A key vehicle for capturing correlation among a discrete set of candidate actions will be Gaussian process regression [5], a method that has met with recent success in predicting outcomes within unknown regions of robot action [6] and observation [7] spaces. An example of the regression performed in a single decision-making step is illustrated in Figure 1. The output of this regression is used to select the maximally informative sensing action from a continuous, local region of the robot's action space. Support vector regression [8] will also be investigated to determine whether similar results can be obtained with reduced computational effort.

1.1 Related Work

Among the earliest information-theoretic exploration strategies are those proposed by Whaite and Ferrie [9] and Elfes [2]. The former work proposes exploring an a priori unknown environment with the goal of minimizing entropy, and the latter work specifically proposes exploring to maximize the mutual information between

sensor observations and an occupancy grid map. More recent works in information-theoretic exploration have considered the trade-off between maximizing mutual information and managing the localization uncertainty in a robot's simultaneous localization and mapping (SLAM) process [10], [11], [12], in addition to the selection of trajectories that maximize map accuracy [13]. Efforts to reduce the computational cost of evaluating mutual information over many possible future measurements have considered small, carefully selected sets of candidate trajectories, using a skeletonization of the known occupancy map [14] and the evaluation of information gain over a finite number of motion primitives [15], [16] or 3D viewpoints [17]. Limiting consideration to local neighborhoods of configurations permits efficient exploration by manipulators in 3D environments [18].

Gaussian process regression has been applied to the problem of robot action selection and control in a variety of contexts. It has been used to solve optimal control problems [6], [19], generate paths that reduce localization uncertainty [20], and select actions that are likely to observe physical objects of interest [21]. It has also been used to aid the inspection of structures by predicting the regions of variability in greatest need of additional measurement [22]. Gaussian process regression has also been used to generate maps that support the evaluation of informative actions [15], [23]. However, to the best of our knowledge, it has not been applied to the problem of action selection for the exploration of unknown environments modeled by occupancy maps, nor has support vector regression.

1.2 Paper Organization

We propose and describe below a methodology for choosing the most informative action from the continuous action space with the aid of supervised learning. A formal definition of the problem is given in Section 2, including brief introductions to Gaussian process regression and support vector regression. The proposed algorithmn is given in Section 3, and the time complexity of the algorithm is analyzed in Section 4. Computational results are presented in Section 5, with conclusions and a discussion of areas for future work in Section 6.

2 Problem Definition

2.1 Information Gain

We define the space of mobile robot sensing actions to be the configuration space $\mathscr{C} \subseteq \mathbb{R}^d$, a subset of d-dimensional Euclidean space. We assume the robot's range sensor provides a 360-degree field of view, and that its occupancy grid map is discretized finely enough to represent the configuration space, in addition to serving

as the robot's model of the environment. In the absence of obstacles, the robot is assumed capable of travel from any grid cell in the map to any other cell. A fundamental presumption in this formulation is that the robot's action space is a subset of the spatial configuration space; this, along with our other assumptions, are similar to those made in [4]. The implications of extending the proposed method to systems with more challenging topologies will be discussed in Section 6.

We define Shannon's entropy [24] over an occupancy grid map m as follows:

$$H(m) = -\sum_i \sum_j p(m_{i,j}) \log p(m_{i,j}) \tag{1}$$

where index i refers to the individual grid cells of the map and index j refers to the possible outcomes of the Bernoulli random variable that represents each grid cell, which is either free or occupied. Cells whose contents have never been observed are characterized as $p(m_{i,j}) = 0.5$, contributing one unit of entropy per cell. Cells whose contents are perfectly known contribute no entropy to the summation.

We use mutual information $I(m,x_i)$ to evaluate the expected information gain with respect to a specific configuration x_i, defined as follows:

$$I(m,x_i) = H(m) - H(m|x_i) \tag{2}$$

where $H(m)$ is the current entropy of the map, and $H(m|x_i)$ is the expected entropy of the map given a new sensor observation at configuration x_i. Our goal is to pick the optimal configuration x^* that maximizes the expected information gain.

$$x^* = \underset{x_i \in \mathscr{C}_{action}}{\operatorname{argmax}} \; I(m,x_i) \tag{3}$$

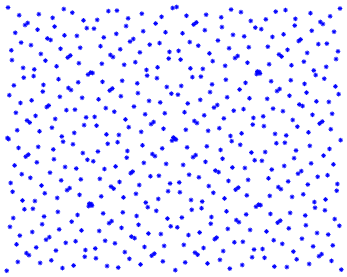
In (3), $\mathscr{C}_{action}$ represents the subset of the configuration space from which the robot's next sensing action will be selected, typically within a short distance of the robot's current location.

2.2 Gaussian Process Regression

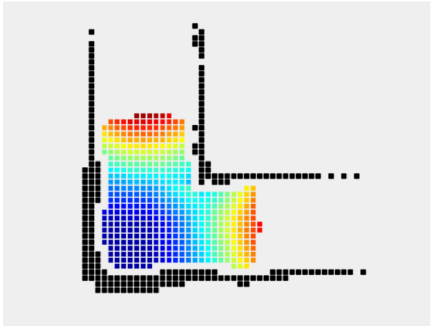
We assume a set of training data $\mathbf{x}$ represents the candidate sensing configurations x_i for which $I(m,x_i)$ has been computed. The values of $I(m,x_i)$ for all $x_i \in \mathscr{C}_{action}$ comprise the set of training outputs $\mathbf{y}$. Gaussian process regression [5] estimates the output values and corresponding covariance associated with a set of test configurations $\mathbf{x}_*$, according to Equations (4) and (5). The test configurations $\mathbf{x}_*$ will be finely discretized, with the same resolution as the occupancy grid map.

$$\bar{\mathbf{y}}_* = k(\mathbf{x}_*,\mathbf{x})[k(\mathbf{x},\mathbf{x}) + {\sigma_n}^2 I]^{-1}\mathbf{y} \tag{4}$$

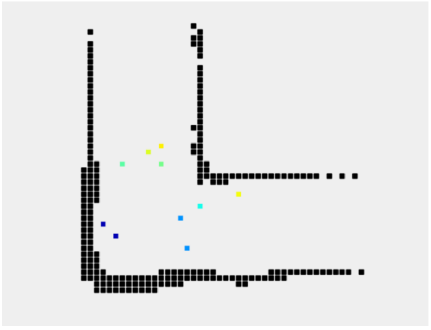
$$cov(\mathbf{y}_*) = k(\mathbf{x}_*,\mathbf{x}_*) - k(\mathbf{x}_*,\mathbf{x})[k(\mathbf{x},\mathbf{x}) + {\sigma_n}^2 I]^{-1} k(\mathbf{x},\mathbf{x}_*) \tag{5}$$



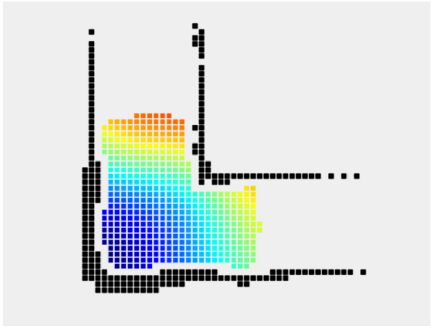
(a) Two-dimensional Sobol sequence.



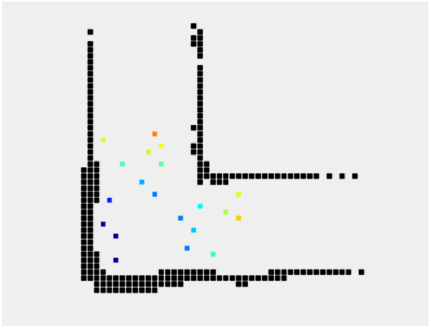
(b) MI evaluated at all grid cells in $\mathscr{C}_{action}$.



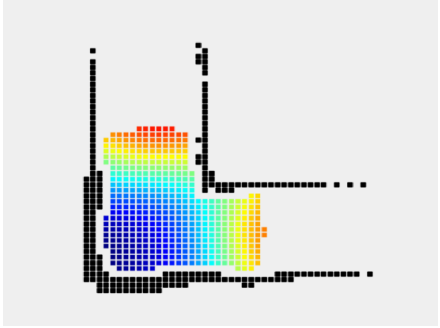
(c) 10 Sobol samples in $\mathscr{C}_{action}$.



(d) GP regression from training data of (c).



(e) 20 Sobol samples in $\mathscr{C}_{action}$.



(f) GP regression from training data of (e).

Fig. 2: Gaussian process (GP) regression using samples from the Sobol sequence. In all images, the robot's current location is the same as depicted in Figure 1, and the black cells represent known obstacles in the robot's occupancy grid. The color scale from blue to red indicates increasing mutual information (MI).

In the above equations, $\bar{\mathbf{y}}_*$ are the estimated values $I(m, x_{i*})$ for the test data $\mathbf{x}_*$, $cov(\mathbf{y}_*)$ is the covariance associated with these outputs, σ_n^2 is a vector of Gaussian noise variances associated with the observed outputs $\mathbf{y}$, and k($\mathbf{x}$,$\mathbf{x}$') is the kernel

function, which gives a covariance matrix relating all pairs of inputs. The hyperparameters of the kernel function, which typically influence such characteristics as smoothness and length scales, can be trained using a preliminary set of representative training data.

We adopt a Matérn kernel function for this application, given by (6).

$$k(\mathbf{x},\mathbf{x}') = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}|\mathbf{x}-\mathbf{x}'|}{\ell}\right)^{\nu} K_{\nu}\left(\frac{\sqrt{2\nu}|\mathbf{x}-\mathbf{x}'|}{\ell}\right) \tag{6}$$

In (6), ν is a parameter used to vary the smoothness of the covariance, ℓ is a characteristic length, Γ is the gamma function, and K_{ν} is a modified Bessel function. In contrast with the squared exponential kernel function, which is more commonly used in Gaussian process regression [5], the Matérn kernel can be tuned to capture sharp variations in the estimated outputs. This has met with success in Gaussian process occupancy mapping, in which sharp and sudden transitions in occupancy probability due to obstacles are successfully modeled [23], [25]. Similarly, we anticipate sharp variations in mutual information due to the presence of obstacles, which will obstruct the visibility of some areas and permit the observation of others.

Example regressions performed over two sets of candidate training data are given in Figure 2. The training data $\mathbf{x}$ is sampled from the two-dimensional Sobol sequence [26]. This pseudorandom sequence is selected to impose some regularity on the training data, as demonstrated by the field of samples shown in Subfigure (a). Subfigures (b), (c), and (e) represent the explicit evaluation of expected mutual information over several series of candidate views. Subfigures (d) and (f) represent the use of Gaussian process regression to predict the mutual information achieved by all actions within the finely discretized grid representing $\mathscr{C}_{action}$, the continuous action space. The boundaries of $\mathscr{C}_{action}$ are set according to the limits of the robot's field of view at its present location.

2.3 Support Vector Regression

Support vector regression [8] is an adaptation of the support vector machine [27] used for regression rather than classification. With the same training set $\mathbf{x}$ of candidate robot configurations employed as training inputs, and $I(m,x_i)$ for all $x_i \in \mathscr{C}_{action}$ as the set of training outputs $\mathbf{y}$, support vector regression estimates the output values associated with test configurations $\mathbf{x}_*$, but not their corresponding covariance, in contrast to Gaussian process regression.

Specifically, we adopt ε-support vector regression, in which we aim to find an approximate function with deviation less than ε from each output value at training time. The basic form for the hypothesis of a support vector regression estimator for test data $\mathbf{x}_*$ is as follows:

$$\bar{\mathbf{y}}_* = \sum_{i=1}^{l} (-\alpha_i + \alpha_i{}^*) k(\mathbf{x}_{*i}, \mathbf{x}) + b \tag{7}$$

where $k(\cdot)$ denotes the Matérn kernel function mapping from test inputs to features and $\alpha_i{}^* - \alpha_i$ denotes the learned weight for the i-th feature in $k(\cdot)$, from the solution to the dual problem described in [28]. Using this approach, results visually indistinguishable from those in Figures 1 and 2 were obtained. This was achieved with reduced computational effort, with the only drawback being the lack of the empirical confidence in the results provided by the covariance of the test data, which would be supplied by Gaussian process regression. This method will be examined alongside Gaussian process regression to determine whether comparable results can be obtained with reduced computational effort in all problems of interest.

3 Algorithm Description

Algorithm 1 AutonomousExploration(x_{init}, m_{init}, InfoThreshold, $N_{samples}$)

$x_k \leftarrow x_{init}$; $m_k \leftarrow m_{init}$; $ActionHistory \leftarrow x_{init}$;
for $k \in \{1, 2, ..., NumIterations\}$ **do**
  $ActionSet \leftarrow \emptyset$;
  $MISet \leftarrow \emptyset$;
  for $x_i \in \mathscr{C}_{action}(x_k, N_{samples})$ **do**
    $MI \leftarrow ObservationPrediction(x_i, m_k)$
    $MISet \leftarrow MISet \cup MI$;
    if $MI > InfoThreshold$ **then**
      $ActionSet \leftarrow ActionSet \cup x_i$;
    end if
  end for
  $ActionSet \leftarrow Regression(ActionSet, InfoThreshold, MISet, x_k, m_k)$;
  if $ActionSet \neq \emptyset$ **then**
    $x_{k+1} \leftarrow BestAction(ActionSet)$;
    $ActionHistory \leftarrow ActionHistory \cup x_{k+1}$;
  else
    $x_{k+1} \leftarrow ActionHistory(PreviousAction)$;
    $ActionHistory \leftarrow ActionHistory \setminus x_k$;
  end if
end for
$m_{k+1} \leftarrow MapUpdate(x_{k+1})$;

The exploration process proceeds according to Algorithm 1. On each iteration, an action set $\mathscr{C}_{action}$ is formulated within the sensor field of view at the robot's current location, and a designated number of sampled actions within the set is evaluated per Equation (2), drawn from a Sobol sequence. Actions whose mutual information surpasses a designated threshold $InfoThreshold$ are added to a set of approved candidate actions $ActionSet$. All of the mutual information data are then used to

Algorithm 2 MI = ObservationPrediction(x_i, m_i)

$m \leftarrow m_i$;
for $beam_j \in SensorBeams(x_i)$ **do**
  $IntersectCell \leftarrow IntersectionDetected(beam_j, m)$;
  if $IntersectCell \neq \emptyset$ **then**
    $r_j = knnsearch(x_i, IntersectCell)$;
  else
    $r_j = MaxSensorRange$;
  end if
  $m \leftarrow EntropyUpdate(x_i, r_j)$;
end for
$MI = Entropy(m_i) - Entropy(m)$;
return MI;

perform the regression of choice to estimate the information gain of all other members of $\mathscr{C}_{action}$ whose mutual informtion was not explicity computed. Actions whose estimated mutual information exceeds the $InfoThreshold$ are also added to the set of candidate actions $ActionSet$. If at least one action is identified whose information gain surpasses the threshold, the robot performs the maximally informative sensing action. However, if none of the actions evaluated surpasses the threshold, the robot takes a step backwards and considers the actions at a previous location along the route traveled, where there may have been informative candidate actions that were not yet performed. The algorithm repeats until the designated number of iterations is performed, or map entropy drops below a user-designated lower limit.

Algorithm 2 gives the specific steps required to explicitly evaluate the mutual information at a designated sample action x_i. This entails a ray tracing computation along each of the robot's sensor beams, returning the ranges to the nearest obstacles intersected, if any. New entropy values are estimated in all cells that are anticipated to be intersected by a sensor beam, and the new expected map entropy is used to compute the expected mutual information after performing the designated sensing action. Algorithm 2 is a sub-routine of Algorithm 1 used to evaluate every Sobol sample from the action space whose mutual information is explicitly computed.

4 Analysis

When using Gaussian process regression, the computational complexity of Algorithm 1 is given in (8):

$$O(N_{steps}(N_{samples}N_{beams}N_{cells} + N_{samples}^3 + N_{samples}^2 N_{actions})) \tag{8}$$

where N_{steps} is the total number of sensing actions taken by the robot in the course of exploration, $N_{samples}$ is the number of designated configurations whose mutual information is explicitly evaluated, $N_{actions}$ is the total number of actions comprising $\mathscr{C}_{action}$ that are estimated using Gaussian process regression, N_{beams}

is the number of beams emitted by the robot's range sensor, and N_{cells} is the worst-case number of occupancy grid cells that a beam may intersect. The term $N_{samples}N_{beams}N_{cells}$ represents the cost of explicitly evaluating mutual information in all cells intersected by the robot's sensor, for all designated actions $N_{samples}$. The term $N_{samples}^3 + N_{samples}^2 N_{actions}$ represents the cost of performing the subsequent Gaussian process regression, which requires the inversion of a matrix that is square in $N_{samples}$, and its subsequent multiplication with cross-covariance terms that scale with $N_{actions}$, the total number of sensing actions recovered from the "test data" of the Gaussian process regression. In practice, we have worked with $10 \leq N_{samples} \leq 20$, $N_{actions} \sim 300$, $N_{beams} = 360$, and $N_{cells} \sim 25$, and we have found that in this range, the complexity of the procedure is dominated by the first term, with the cost of the Gaussian process regression relatively minor in comparison to the cost of the mutual information computation. Hence, a much larger number of sensing actions can be evaluated approximately for a small additional cost on top of the initial evaluation of information gain over the original set of samples. Specific examples will be highlighted in the following section.

When using support vector regression, the complexity of Algorithm 1 is given in (9):

$$O(N_{steps}(N_{samples}N_{beams}N_{cells} + N_{samples}^3 + N_{actions})) \tag{9}$$

where training still takes on worst-case cubic complexity (which occurs when the upper bound on the coefficients α_i is large) but testing is linear in the number of candidate sensing actions. Once again, the complexity of the procedure is dominated by the $N_{samples}N_{beams}N_{cells}$ term, and the cost of the regression is minor in comparison to the cost of mutual information computation.

5 Computational Results

5.1 Experimental Setup

We explored the performance of our algorithm using two different maps: 1) a "maze" map representing an indoor environment (shown in Figure 3) and 2) an "unstructured" map representing a forest-like environment (shown in Figure 4). In our simulations, we assume the robot is equipped with a laser scanner with a 360° field of view and 1° resolution. The range of the laser scanner was set to 1 meter, and all sensing actions considered were within a 0.5m range of the robot, ensuring that the next sensing action lies within the robot's current field of view to the extent that its outcome can be reasonably predicted by a mutual information evaluation over the existing map. The exploration process was simulated using MATLAB.

We initialized the robot randomly within each map and simulated 100 instances of exploration for each of the six following cases: a) choosing the best action among 10 Sobol samples (as depicted in Figure 2(c)), b) choosing the best action among

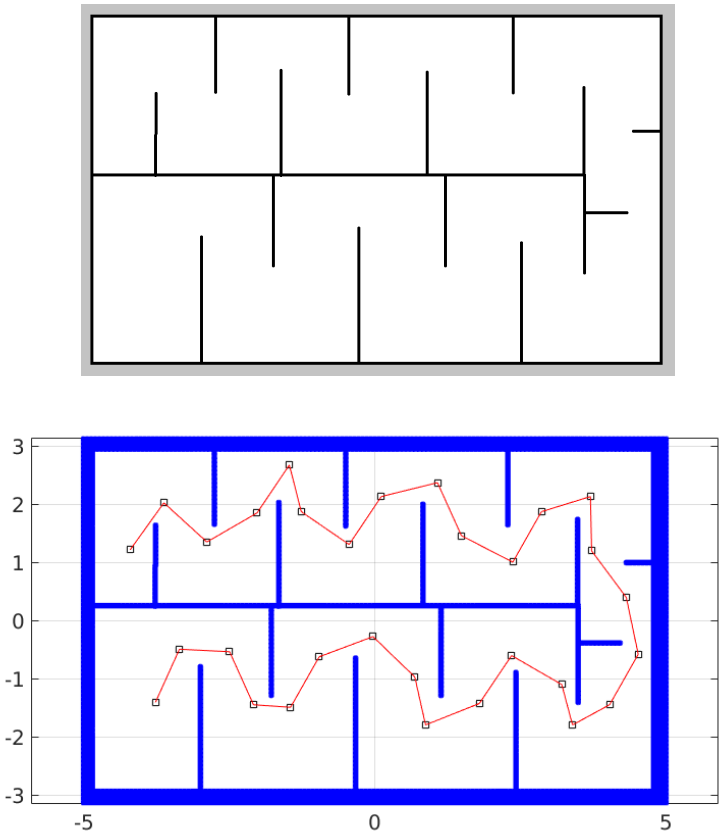


Fig. 3: The "maze" map used in our experiments is shown at top. The dimensions of the map are approximately 6 by 9 meters. An occupancy map produced by exploration with GP regression, showing the trajectory of the robot, is given at bottom.

20 Sobol samples (as depicted in Figure 2(e)), c) using 10 Sobol samples as the basis for Gaussian process regression and d) support vector regression, then choosing the best action from the approximately continuous action space, and e) using 20 Sobol samples as the basis for Gaussian process regression and f) support vector regression, again choosing the best action from the approximately continuous action space. The robot is permitted to explore until its map entropy falls below a designated threshold, after which the simulation terminates. The Gaussian process regression computations were performed with the aid of the Gaussian processes for machine learning (GPML) MATLAB library [29], and support vector regression computations were performed in MATLAB using LIBSVM [28], with a precomputed Matérn kernel. The computation required for each trial was distributed across

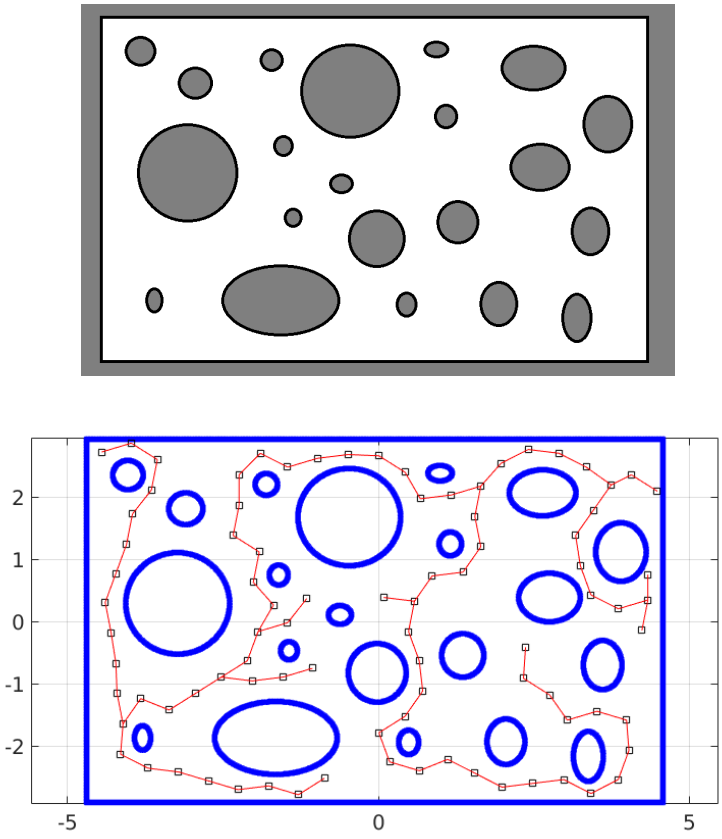
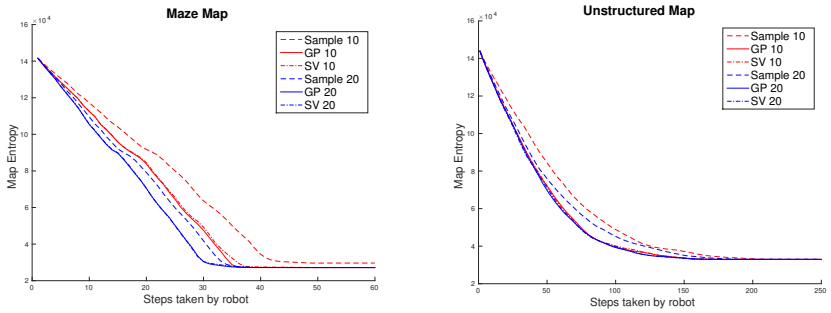


Fig. 4: The "unstructured" map used in our experiments is shown at top. The dimensions of the map are approximately 7 by 9 meters. An occupancy map produced by exploration with GP regression, showing the trajectory of the robot, is given at bottom.

four cores of an Intel Xeon 5 3.0 GHz processor using the MATLAB Parallel Computing Toolbox, and a computer equipped with 4GB RAM.

5.2 Results

As noted in Section 4, the time consumed by the regression computations across the entirely of the action space is substantially less than evaluating the mutual information across the much smaller designated set of actions drawn from a Sobol sequence. Figure 5 gives results showing the performance of the six problem pa-



(a) Results from the maze map of Fig. 3.

(b) Results from the unstructured map of Fig. 4.

Fig. 5: The results of 100 exploration trials randomly initialized in their respective maps, for each of six parameterizations. The mean entropy reduction is given over the number of sensing actions performed by the robot for all test cases considered.

rameterizations over the maps of Figures 3 and 4. In the maze map, both supervised learning methods drive down entropy faster than each respective case that chooses the most informative action from the explicitly evaluated sample set. In this case, all exploration methods nearly always select sensing actions from the same homotopy class, but choosing the approximately continuous action that is expected to be most informative, with the aid of supervised learning, tends to point the robot in a more advantageous direction than the most informative Sobol sample.

In the unstructured map, all parameterizations using Gaussian process and support vector regression perform better across the board, even when less computational effort is invested in establishing a training data set. In this case, the learning-based methods occasionally select actions from a different homotopy class than the competing method using explicitly computed mutual information only, resulting in fundamentally different paths among the different parameterizations. The use of supervised learning to select moves from the continuous space of sensing actions accumulates a more significant advantage, such that regression over 10 samples performs better than explicity evaluating the mutual information at 20 samples. Hence, more informative outcomes are selected with substantially less computational effort. Representative trajectories of the robot when using Gaussian process regression over 20 Sobol samples are given in Figures 3 and 4, for each map. These trajectories represent full exploration of their respective environments, reaching the lower allowed limit on map entropy. Figure 6 gives representative examples of different exploration outcomes resulting from Gaussian process exploration, versus exploration using the sampled configurations only. Finally, Table 1 gives details on the computation time required, and the number of steps taken by the robot in the exploration process, for all examples implemented over the unstructured map of Figure 4.

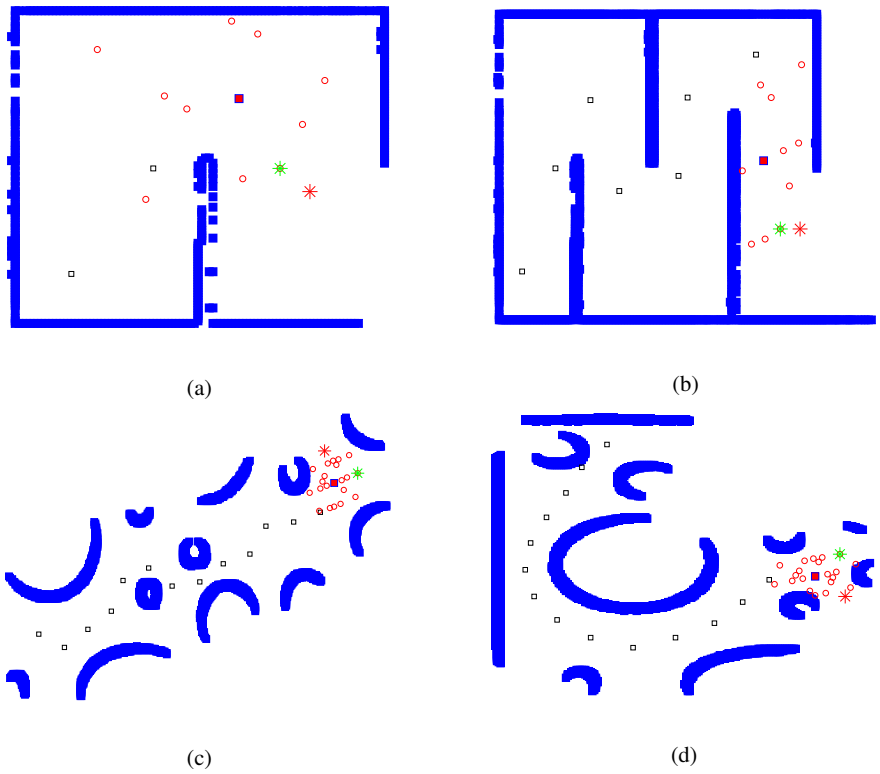


Fig. 6: Representative cases when Gaussian Process regression makes a more informative decision than the best sample whose information gain was explicitly evaluated. The green star represents the most informative Sobol sample, and the red star represents the action expected to be most informative from the Gaussian process regression. In cases (c) and (d) from the unstructured map, the candidate actions lie in different homotopy classes.

6 Conclusions and Future Work

We have proposed a novel approach to evaluate the mutual information throughout a robot's continuous action space, for the purpose of exploring a priori unknown environments. In the examples considered, supervised learning facilitates the selection of more informative sensing actions, in some cases selecting more informative actions with less overall computational effort.

Time Cons. Per Step (Secs)	**10 GP**	**10 SV**	**10 DA**	**20 GP**	**20 SV**	**20 DA**
μ	2.18	2.18	2.19	3.83	3.79	3.79
σ	0.03	0.02	0.03	0.04	0.04	0.04
Steps Taken Per Trial	**10 GP**	**10 SV**	**10 DA**	**20 GP**	**20 SV**	**20 DA**
μ	177.2	177.4	222.7	175.3	174.9	199.9
σ	4.87	5.06	5.65	5.11	5.14	5.58

Table 1: The results shown here are the average of 100 computational trials over the unstructured map shown in Figure 4. For the six test cases examined (in which "DA" refers to the deterministic approach, derived from Sobol samples, "GP" refers to the Gaussian process approach, and "SV" refers to the support vector approach), at top we show a comparison of the mean and standard deviation of computation time required per sensing action, and at bottom we show a comparison of the mean and standard deviation of the total number of steps taken by the robot in the course of driving its entropy to the minimum designated value.

6.1 Complex Action Spaces

Extending the approach of this paper to higher-dimensional systems and non-Euclidean action spaces is a compelling area for future work, in which we intend to consider regression over a robot's rotational degrees of freedom, as well as for differential systems capable of aggressively exploring their environments. The Matèrn kernel function shows promise in capturing sharp variations in expected mutual information across obstacle boundaries, and we intend to test its applicability to even more sharply varying action spaces in future work.

Acknowledgements The authors would like to thank Chengjiang Long from the Computer Science Department and Kiril Manchevski from Mechanical Engineering Department of Stevens Institute of Technology for help with computational resources.

References

1. S. Thrun, W. Burgard, and D. Fox, "Exploration," In *Probabilistic Robotics*, pp. 569-605, MIT Press, 2005.
2. A. Elfes, "Robot Navigation: Integrating Perception, Environmental Constraints and Task Execution within a Probabilistic Framework," *Proceedings of the International Workshop on Reasoning with Uncertainty in Robotics*, pp. 93-129, 1995.
3. A. Elfes, "Using Occupancy Grids for Mobile Robot Perception and Navigation," *Computer*, vol. 22(6), pp. 46-57, 1989.

4. B.J. Julian, S. Karaman, and D. Rus. "On Mutual Information-Based Control of Range Sensing Robots for Mapping Applications," *The International Journal of Robotics Research*, vol. 33(10), pp. 1375-1392, 2014.
5. C.E. Rasmussen and C.K.I. Williams, "Gaussian Process for Machine Learning," MIT Press, 2006.
6. M.P. Deisenroth, D. Fox, and C.E. Rasmussen, "Gaussian Processes for Data-Efficient Learning in Robotics and Control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37(2), pp. 408-423, 2015.
7. S.T. O'Callaghan and F.T. Ramos, "Gaussian Process Occupancy Maps," *The International Journal of Robotics Research*, vol. 31(1), pp. 42-62, 2012.
8. A. Smola and B. Schlkopf. "A tutorial on support vector regression." *Statistics and Computing*, vol. 14(3), pp. 199-222, 2004.
9. P. Whaite and F.P. Ferrie, "Autonomous Exploration: Driven by Uncertainty," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19(3), pp. 193-205, 1997.
10. F. Bourgault, A.A. Makarenko, S.B. Williams, B. Grocholsky, and H.F. Durrant-Whyte, "Information Based Adaptive Robotic Exploration," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 540-545, 2002.
11. C. Stachniss, G. Grisetti, and W. Burgard, "Information Gain-Based Exploration Using Rao-Blackwellized Particle Filters," *Proceedings of the Robotics: Science and Systems Conference*, pp. 65-72, 2005.
12. A. Kim and R.M. Eustice, "Perception-Driven Navigation: Active Visual SLAM for Robotic Area Coverage," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3196-3203, 2013.
13. T. Kollar and N. Roy, "Trajectory Optimization using Reinforcement Learning for Map Exploration," *The International Journal of Robotics Research*, vol. 27(2), pp. 175-196, 2008.
14. T. Kollar and N. Roy, "Efficient Optimization of Information-Theoretic Exploration in SLAM," *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 1369-1375, 2008.
15. K. Yang, S.K. Gan, and S. Sukkarieh, "A Gaussian Process-Based RRT Planner for the Exploration of Unknown and Cluttered Environment with a UAV," *Advanced Robotics*, vol. 27(6), pp. 431-443, 2013.
16. B. Charrow, V. Kumar, and N. Michael, "Approximate Representations for Multi-Robot Control Policies that Maximize Mutual Information," *Autonomous Robots*, vol. 37(4), pp. 383-400, 2014.
17. K.M. Wurm, D. Hennes, D. Holz, R.B. Rusu, C. Stachniss, K. Konolige, and W. Burgard, "Hierarchies of Octrees for Efficient 3D Mapping," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4249-4255, 2011.
18. P. Quin, G. Paul, A. Alempijevic, D. Liu, and G. Dissanayake, "Efficient Neighbourhood-Based Information Gain Approach for Exploration of Complex 3D Environments," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1343-1348, 2013.
19. M.P. Deisenroth, C.E. Rasmussen, and J. Peters, "Gaussian Process Dynamic Programming," *Neurocomputing*, vol. 72(7-9), pp. 1508-1524, 2009.
20. R. Martinez-Cantin, N. de Freitas, A. Doucet, and J.A. Castellanos, "Active Policy Learning for Robot Planning and Exploration Under Uncertainty," *Proceedings of the Robotics: Science and Systems Conference*, 2007.
21. J. Velez, G. Hemann, A.S. Huang, I. Posner, and N. Roy, "Modelling Observation Correlations for Active Exploration and Robust Object Detection," *Journal of Artificial Intelligence Research*, vol. 44, pp. 423-453, 2012.
22. G.A. Hollinger, B. Englot, F.S. Hover, U. Mitra, and G.S. Sukhatme, "Active Planning for Underwater Inspection and the Benefit of Adaptivity," *The International Journal of Robotics Research*, vol. 32(1), pp. 3-18, 2013.
23. M.G. Jadidi, J.V. Miro, R. Valencia, and J. Andrade-Cetto, "Exploration on Continuous Gaussian Process Frontier Maps," *Proceedgins of the IEEE International Conference on Robotics and Automation*, pp. 6077-6082, 2014.
24. C.E. Shannon and W. Weaver, *The Mathematical Theory of Communication*, University of Illinois Press, 1949.

25. S. Kim and J. Kim, "Continuous Occupancy Maps using Overlapping Local Gaussian Processes," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4709-4714, 2013.
26. I.M. Sobol, "On the Distribution of Points in a Cube and the Approximate Evaluation of Integrals," *USSR Computational Mathematics and Mathematical Physics*, vol. 7(4), pp. 86-112, 1967.
27. C. Cortes and V. Vapnik, "Support-vector Networks," *Machine Learning*, vol. 20(3), pp. 273-297, 1995.
28. C. Chang and C. Lin, "LIBSVM : A library for support vector machines." *ACM Transactions on Intelligent Systems and Technology*, vol. 2(3), 2011.
29. C.E. Rasmussen and H. Nickisch, "Gaussian Processes for Machine Learning (GPML) Toolbox," *The Journal of Machine Learning Research*, vol. 11, pp. 3011-3015, 2010.